MACHINE VISION FOR THE DETERMINATION OF IDENTITY, ORIENTATION AND POSITION OF TWO DIMENSIONAL INDUSTRIAL COMPONENTS.

Jonathan English

A thesis submitted in partial fulfilment of the requirements of De Montfort University for the degree of Doctor of Philosophy

March 1996

ABSTRACT

The determination of identity, orientation and position for two dimensional shapes is common to many industrial problems. The work presented in this thesis investigates a machine vision solution for one such industry, namely that of shoe manufacture. There are no constraints on the design of shoe components which results in a vast assortment of shapes in various sizes. Thus, it is anticipated that the results gleaned from this work will be transferable to other industries dealing in two dimensional shape recognition.

To facilitate identification the components, which are of a flexible nature, are scanned using a linescan camera and converted into a binary silhouette representation. The silhouette is used to construct a feature description of the component, this is then used to determine its identity, orientation and position. In theory the recognition process should be deterministic but due to production and imaging system variations this is not the case. Certain separate components differ by just a few percent and the recognition system must promise industrially acceptable speed so the selection of a suitable feature representation is critical.

One process within the shoe industry is that of stitching decorative patterns onto shoe upper components prior to assembly. In the past machine vision based on the description of shapes using a centroid to boundary radii method has been used to automate this process. Investigations into an alternative and improved methodology, based on moment invariants, are presented here.

Moments and moment invariants were researched to investigate their performance with a view to improving the existing method and produced some promising results. They combined very well with the line scan imaging system to give a feature description that was economical in storage and efficiently extracted. The basic elements of the feature set provided all the information required to identify the shape as well as to locate its position and assess its orientation. Despite the problems of boundary noise associated with the higher order moments the recognition success rate was as good as that of the radii system.

Experiments on identical images were used to evaluate methods for improving the recognition performance of the moment invariant feature vector. The results show that the use of noise filtering, correcting for grid errors and neural networks can all reduce feature variation and improve recognition.

Finally a system based on the use of moments is compared quantitatively and contrasted with that using a polar vector method.

DECLARATION

The work contained in this thesis has not been submitted elsewhere for any other degree or qualification and unless otherwise referenced it is the author's own work.

Copyright © 1995 by Jonathan P. English

The copyright of this thesis rests with the author. No quotation from it should be published without J. P. English's prior written consent and information derived from it should be acknowledged.

ACKNOWLEDGEMENTS

This research project was jointly funded by the EPSRC - Engineering and Physical Sciences Research Council (formerly the SERC - Science and Engineering Research Council) under a CASE studentship and BUSM Ltd. - British United Shoe Machinery Ltd.

The project was executed under the supervision of Dr. Len Norton-Wayne at De-Montfort University and Mr. David Reedman at BUSM. The author would also like to thank Mr. Hugh Sasse of De-Montfort University for his help with computer system problems, Mr. Arnold Shutt of BUSM for his work with the original hardware and Dr. Nigel Tout of BUSM for his valued co-operation.

CONTENTS

CHAPTER 1 - INTRODUCTION

CHAPTER 2 - BUSM'S EXISTING SYSTEM AND PROPOSED HARDWARE DEVELOPMENTS

CHAPTER 4 - SHAPE REPRESENTATION

CHAPTER 5 - METHODS OF SHAPE DESCRIPTION AND THEIR USEFULNESS FOR DETERMINING IDENTIFICATION, ORIENTATION AND POSITION

CHAPTER 1

INTRODUCTION

1.1 Introduction

Shoe manufacture is a high volume, low cost industry with manufacturers operating in a highly competitive market. Recent years have seen Far Eastern companies take an increased share of the world market due to reduced labour costs. To increase competitiveness companies must reduce the manufacturing cost of their product. British United Shoe Machinery (BUSM) has been developing several machines that will further automate the shoe manufacturing process and reduce the product cost by increasing productivity, maintaining higher quality, reducing wastage and giving the production line more flexibility. The machinery for processing the shoes is individual to each type of task but in all cases there is a common requirement to identify, inspect and monitor the position of component parts using machine vision [BROWNE 1986] techniques.

One particular process being automated to a greater level is that of pre-stitching components with decorative patterns prior to assembly of the shoe upper. Currently this task is performed by operators using automatic sewing machines, a selection of patterns and a jig to hold the work piece in a known position. Any error in selection of component-pattern combination or position leads to discarded work or poor quality finished shoes.

BUSM have developed a machine to increase the automation of this task by processing randomly selected and positioned components. It uses machine vision to assess component identity, position and orientation independently of any operator input. This information can then be used to select a pattern from a data base and adjust it to suit the position of the component prior to control of a sewing machine.

This thesis describes the work undertaken to review the machine vision aspect of the stitcher machine and to evaluate any total or partial processes which could increase the efficiency and accuracy of this system and similar systems which view components as two dimensional, binary silhouettes. In parallel with this research BUSM are

developing a new machine vision hardware unit with which any processes need to be compatible.

1.2 The shoe production cycle

Raw material in the form of leather hides, plastics or similar synthetic material is assessed at the start of the production process during a process known as 'clicking'. Clicking is performed by an operator, known as a Clicker, who will assess any material defects and in the case of leather hides identify the patterns of the material texture and grain. Using this information the Clicker will position knives corresponding to shoe components so as to make best use of the material. The components are then cut out by lowering a large press onto the knives (Appendix I shows some examples of shoe component shapes). A consequence of making the best use of the material is that components are generated in random orders and quantities.

The components produced by the Clicker may be processed individually, grouped as identical components or sorted into kits. A kit comprises all the component parts needed to produce the uppers of several (e.g. 12) pairs of shoes for a particular style and size.

At this point in the cycle any decorative preparations such as pre-stitching will be performed since it is much easier to handle 2 dimensional shapes than those which have been moulded into a 3 dimensional foot shape.

Once the components are ready then they can be assembled into uppers. These are the parts of the shoe above the sole that contain the foot. The uppers are usually assembled flat by gluing or stitching neighbouring components together. In order to produce a flat join between overlapping shapes, neighbouring edges are chamfered in a process called skiving.

Parallel to the process of creating the upper is that of creating the sole. The final part of the shoe assembly operation is to bring the two parts together around a foot shaped former known as a last. The uppers are then fixed to the sole by stitching, gluing or a combination of the two. Prior to gluing the shiny material surfaces are scoured by a roughing machine in order to remove the cosmetic finish and give the glue a key.

A guide to the making of shoes can be found in the 'Manual of Shoemaking' edited by Miller [MILLER 1989].

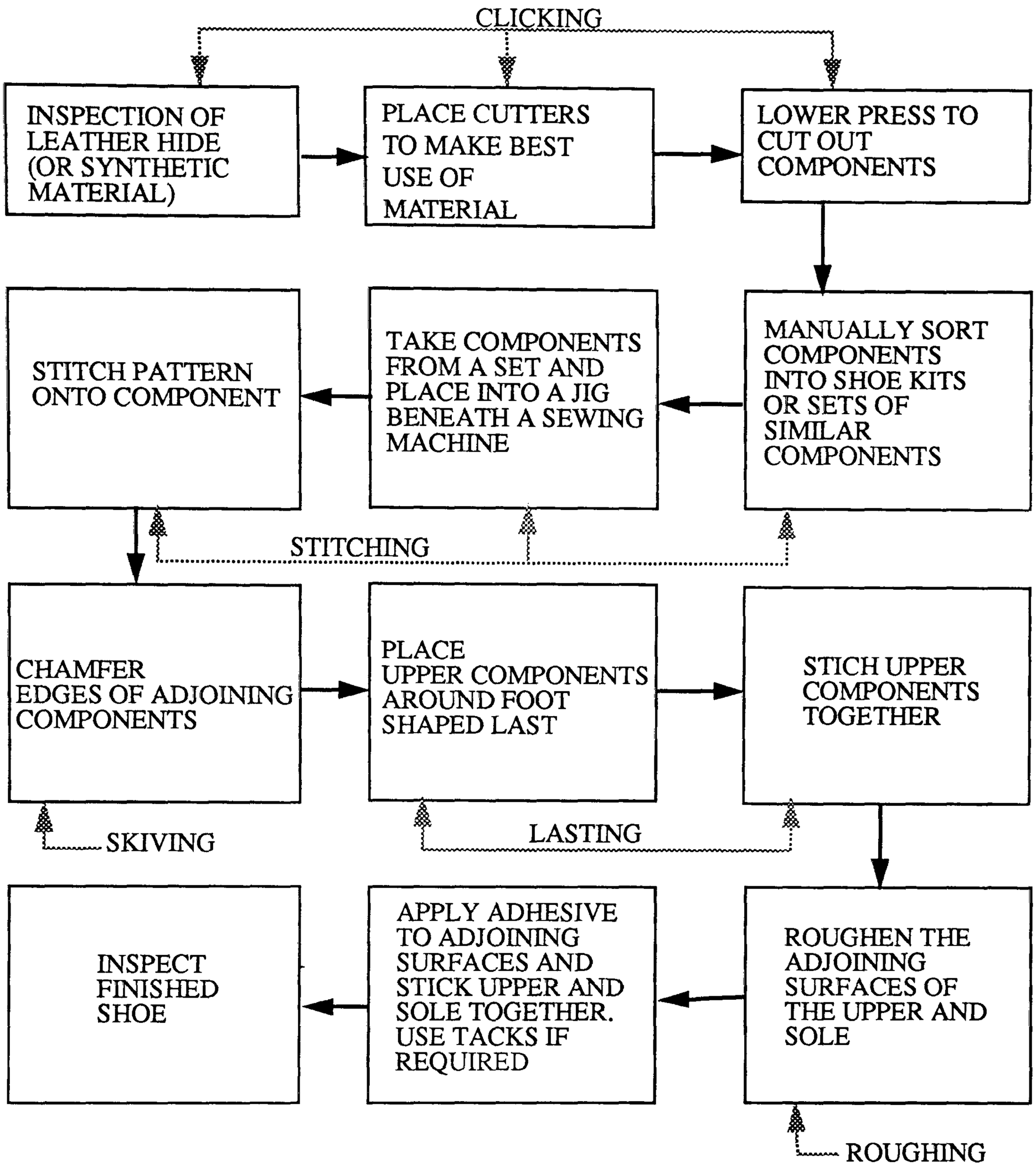


Figure 1.1 Shoe manufacture process.

1.3 Automation

Automation is the use of machines to perform sets of tasks autonomously with little or no human interaction. Automatic machines have been employed in a wide variety of industries but they can be roughly split into either inspection machines, manufacturing machines or a mixture of the two such as systems for the inspection and packing of objects. Further, they are either used to save money or to reduce the risk to human lives.

The automatic stitching machine proposed in connection with this research falls into the third category, that is to say it is an inspection / manufacturing machine designed to give more economic production.

1.3.1 Advantages associated with automatic machines

Automatic machines work endlessly, without tiring of the task set before them. As a consequence of this the quality of the finished work remains constant. This is particularly valid for work of a repetitive nature and effects both manufacturing and inspection operations. In the first case the quality of the finished article is affected. In inspection the diligence of the inspector will vary with the passage of time leading to quality problems with complete assemblies. Both errors result in higher costs and if complete assemblies make their way onto the market place the cost in terms of lost reputation can be enormous.

The employment of automatic manufacturing and/or inspection leads to lower rejection costs, lower losses through poor quality and lower labour costs through higher production per operator.

1.3.2 Disadvantages

An automatic machine is less flexible than a human operator. The processing of small 'one off' batches is best suited to the dexterity of the human due to the effort required in setting up automatic operations. Automatic machines can also involve large capital outlay and the associated risk of investing in new technology. The use of automation may cause discontentment amongst operatives who may refuse to accept new working practices. The installation, training time and prospective breakdown of a system may also cause down time in the production line.

1.3.3 Ethical considerations

In increasing productivity per operator there is obviously scope for reducing the size of the shop floor workforce. However by decreasing costs any company using automation will become a more competitive force in the market place thus protecting large numbers of jobs that might otherwise be lost through company failures. The current level of imports of shoe type products to the U.K. far exceeds that of exports due to much lower labour costs in the Far Eastern countries. The use of automation brings all labour cost down, allowing companies to compete on similar levels. The design,

development and construction of the new machine vision based systems also creates employment in the high technology sector.

1.4 Overview of machine vision

Machine vision is the use of a camera-computer / hardware system to replace the eye-brain processes performed by humans in tasks such as inspection and machine control. The crucial aspect of a machine vision system is the requirement to extract sought after and usable information from a given image.

So far this thesis has considered automation in its general form, that is, machines controlled in some way (typically electronically or mechanically) to perform a repetitive task. Systems such as car manufacturing robots may be open loop, relying on correct positioning of components to perform satisfactorily. Others, in particular inspection machines, must have a closed loop system whereby information about a component can be obtained. This information is provided by sensors measuring quantities such as weight, volume or temperature. In the case of a machine vision unit the information is gathered by a camera and passed on as a series of pixels describing the light distribution of the image scene.

Once gathered the information must be stored, processed and interpreted in order to make some decision. The image interpretation element, as opposed to image generation, distinguishes machine vision from a closed circuit television system as might be used in security monitoring. In a machine vision system the interpretation process is performed electronically typically in software by a microcomputer or in the case of fast systems by dedicated hardware.

Within an automated system machine vision can be used to make decisions on visibly quantifiable factors such as quality, position and identity. This information can then be fed back and used to control other manufacturing processes.

There are many different ways of configuring a machine vision system. The key elements (figure 1.2) are a camera with suitable optics and lighting conditions, image acquisition hardware, image analysis software (or hardware) and an output process. The input to such a system is generally an image of a scene containing the item to be investigated. No additional labels or tags are available to the processor which must make a decision based only on information it can extract from the image. Basic information might be the item's area, number of corners or its perimeter length. The output might be an alarm if a fault is found, a report of the item's identity or the control of a robot arm requiring the item's position.

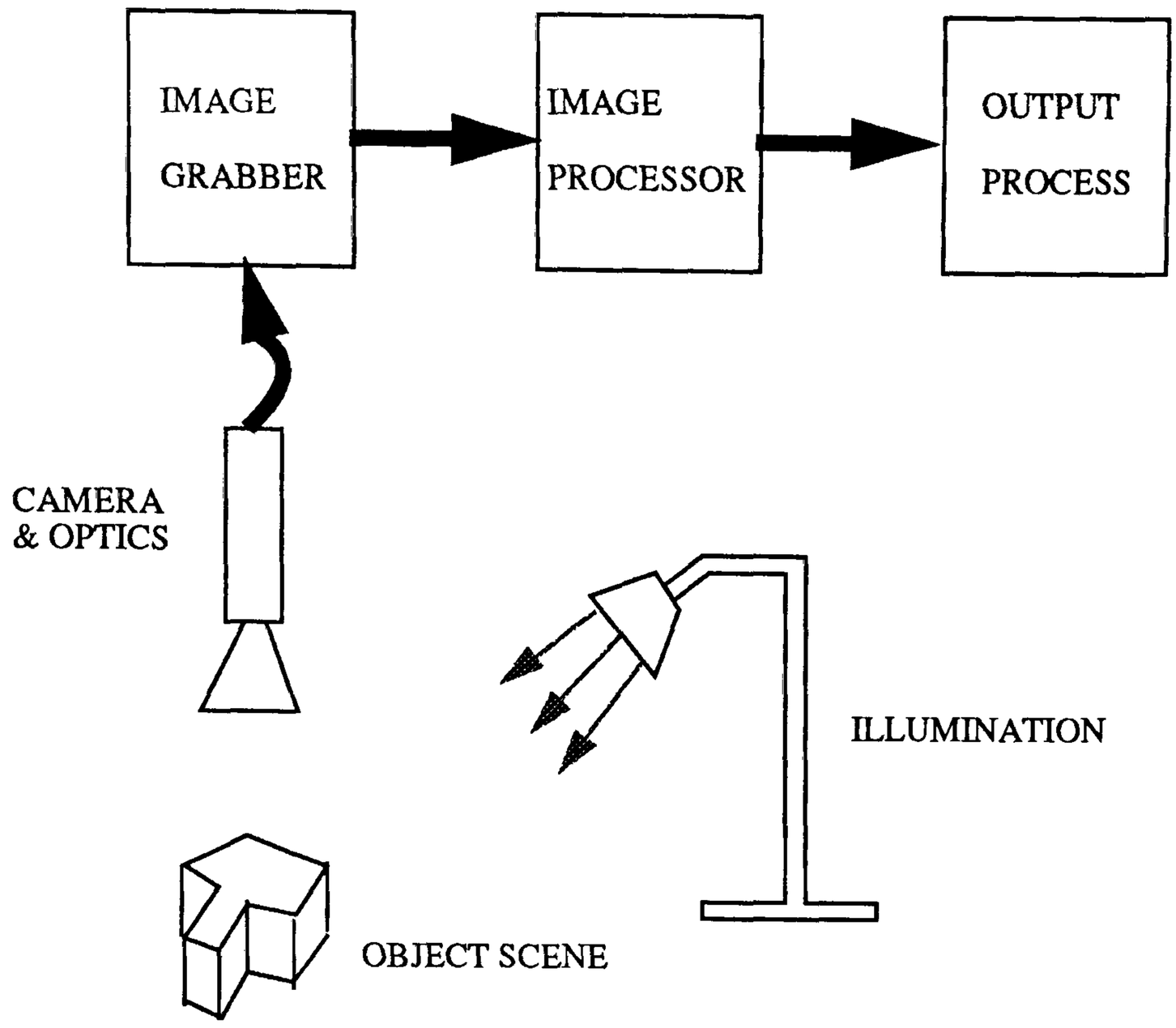


Figure 1.2 An outline machine vision system.

An alternative method available for identification only is to label items with machine readable labels - the bar code and its derivatives [e.g. BURKE 1984][CHARTIER 1994]. Machine vision systems, as described above, can be used to read this type of label but it is more usual to see systems involving laser scanners such as those used at the supermarket check-out. Laser scanners do not require the additional illumination needed by a conventional camera-based system.

Methods which use bar codes obviously require the labels to be attached at some point. This is no problem for packaged items since labels can be pre-printed onto the containers. In the case of shoe components the labels would have to be fixed as part of the clicking process due to the random nature of the operation. Leaving labelling until afterwards would still require a system to identify random, unlabelled components. Once all operations have been completed then the label must either be removable or unnoticeable in the finished shoe.

Some early work on bar code labels for identification of shoe upper components found that conventional 'sticky' labels with sufficient sticking power to withstand the workings of shoe factory machinery left marks on the finished shoe. Similarly labels that did not leave marks were often removed by machinery prior to recognition. Alternative markings such as ultra violet printing survived manufacture and did not

leave a visible mark until the shoes came into contact with an ultra violet light source such as disco lights.

Even if suitable labels can be used they cannot fulfil the requirements to locate a components position and its orientation.

1.5 Potential for machine vision in shoe manufacture

Due to the flexible nature of the materials used in shoe manufacture standard automation techniques involving the use of 'non-intelligent' or open loop machines can not be used. By using machine vision to assess the shoe assembly, machine sequences can be adapted to the particular situation. In addition to this, inspection stages can either be replaced by visual machine inspection or removed altogether due to the consistent high quality performance of a machine.

Applying the technology of machine vision in this way could lead to the automation of the entire shoe factory production line. Figure 1.1 illustrates the shoe production cycle. Current research is aimed at automation of particular sections: clicking, stitching, lasting, roughing and inspection as individual entities. Ultimately all processes might be linked to produce a completely automated manufacturing process from incoming leather hide to outgoing boxed set of shoes.

In addition to the shoe manufacturing tasks listed above there are auxiliary processes which can be automated, these include the making of the cutting knives and the production of the tacks which are sometimes used in shoe assemblies.

1.6 Machine vision in shoe manufacture

In order to benefit from machine vision a problem must involve repeated tasks which can be analysed in order to form a set of rules for operation. There are several processes within the shoe industry which conform to this sort of requirement and have either been developed, are the subject of ongoing research or are possibilities for future research.

1.6.1 Roughing

Roughing is a process applied to two or more leather surfaces prior to the application of an adhesive. It involves scouring the contact areas involved in order to give the adhesive a key. A typical operation such as fixing the completed upper to the sole might require a roughed strip 1cm wide around the outside edge of both the sole and upper

sections. Machine vision techniques could be used to monitor the roughing tool and guide it around the correct course. Such a machine could cope with any style and any size of shoe.

1.6.2 Lasting

The individual two dimensional shoe upper components are joined together in a flat form, usually by sewing, and then moulded around a last to form a 3 dimensional shoe shape. The sole is then attached to the upper either by sewing or gluing. Machine vision could be used to monitor the alignment of the component parts, to control the sewing machine or to control the glue process.

1.6.3 Tack manufacture

The sole assembly of a shoe may make use of tacks, adhesive, thread or any combination of the three. Automatic insertion of tacks has been available for some time but relies on a supply of high quality tacks. Typical tack manufacturing machines produce tacks from wire at the rate of 800 per minute. Previously quality control checks have been performed manually on a sample batch basis. This is expensive both in man power terms and the consequences of faulty tacks getting through the system and damaging customers' expensive machines.

Machine vision offers the opportunity to perform 100% visual inspection. A system has been developed using several cameras to inspect each view of every tack as it comes off the production line. Irregular tacks are automatically discarded leaving only perfect items to be dispatched to the customer [NORTON -WAYNE *et al.* 1985]. A similar system has been developed for inspection of rivets [McCABE *et al.* 1994].

1.6.4 Knife making

The two dimensional shoe components are cut from a sheet of material by a heavy press acting on suitably shaped sections of metal strip known as knives. The knives are formed by bending the knife steel according to a drawing, often generated by a CAD system. Large manufacturers will produce several knives of nominally equivalent shape leading to several different examples of the same shape throughout the production system.

The KARATE [LI 1995] system has been developed to produce knives of perfect shape. It does this by using a camera to acquire images of the knife as it is being bent

and comparing these with the prototype. The shape of the knife being made can then be modified accordingly. This type of approach should produce sets of knives which are identical for any one pattern, thus removing one cause of shoe upper component variation.

1.6.5 Clicking

Clicking is the most complex of the shoe production tasks. It is the job of the 'Clicker' to position the cutting knives that will produce the individual shoe components. The texture, grain, overall shape and any damaged areas of the hide (or synthetic material) must be taken into account so as to make the best use of the available material.

Attempts at the automation of the clicking process have so far been unsuccessful. Determining the positions of the knives so as to maximise the number of components produced is relatively easy. The difficulty lies in assessing the texture and grain of the hide so that each component is of suitable quality for its location in the completed shoe.

1.7 Decorative stitching of components

This research is concerned in the first instance with applying machine vision to the problem of recognising shoe upper components. One application of this is in the pre-stitching of components prior to manufacture. The scheme most widely used at present for decorative component stitching is the jig based method described below.

1.7.1 Jig based method

Components may arrive at the stitching station in one of three forms;

(a) Grouped into nominally identical components requiring identical operations.

(b) Grouped into kits of components required for one pair of shoes of a certain size.

(c) Randomly as produced by the cutting process.

An operator must place each component precisely into a purpose designed jig which in turn must be correctly sited beneath a sewing machine rig. The correct sewing pattern must then be selected from those available. Providing the component has been placed with correct position and orientation, a duplicate of an original pattern will be stitched onto it by the controlling computer. Any failure in positioning or selection will lead to a misaligned or incorrect pattern.

1.7.2 Cost

The capital costs of this type of automatic sewing machine are around £35,000 to £40,000 excluding jigs. The cost of employing machine operators is around £3 per hour per machine.

1.7.3 Productivity rates

The rate at which completed components are produced will depend on the size of components and the complexity of the stitched pattern but on average a practised operator might produce 120 completed parts per hour.

1.7.4 Wastage

If a component is positioned incorrectly or the wrong pattern is selected for stitching then waste will be generated The levels of waste will depend on operator experience and pattern complexity. The direct cost of such waste will be small but the loss of a component may have ramifications further down the production line if 'just in time' production systems are in use.

If errors are not spotted until components are incorporated into more complete units then whole shoes may have to be scrapped with considerably higher costs.

1.7.5 Reputations

If any faulty components should make their way into finished shoes, onto the market place and into the hands of customers then the costs in terms of lost reputation and business could be very large. The result of this is that quality inspectors must be employed at several stages of production.

1.7.6 Machine vision and component pre-stitching

The jig based method lacks flexibility and relies on the operator to identify each component and position it correctly. By incorporating machine vision techniques into the system the component can be automatically identified, its position determined and its orientation calculated. This information can then be used to adjust the control of the sewing machine so that the correct pattern is selected and manipulated to suit the component position. This type of system should be more productive, use fewer

machines (reduce overall capital costs) and produce less waste than the jig based method outlined above, thus reducing manufacturing costs.

1.8 Summary

In order to reduce costs while maintaining quality and production rates the shoe industry, as with many others, is turning to automation. The use of machine vision allied to automatic machines provides an opportunity to further the uses to which automation can be extended by applying new flexibility, in terms of sensory input, to machine control.

This project details research into a machine vision module for identifying 2 dimensional shoe upper components, calculating their position and determining their orientation. The majority of shoe components are produced from leather and it is the properties of the leather (e.g. rough edges, flexibility, softness) along with the undetermined nature of the component design that are particularly challenging for machine vision identification.

Automation is applicable to a wide variety of tasks and due to the difficult nature of shoe component identification it is anticipated that successful methods will be transferable to other industrial problems where components can be represented by two dimensional silhouettes.

CHAPTER 2

BUSM'S EXISTING SYSTEM AND PROPOSED HARDWARE DEVELOPMENTS

2.1 Introduction

British United Shoe Machinery (BUSM) are the world's major manufacturer of shoe making machinery with a turnover of £200 million and a 20% share of the world market. The company sustains this position through continual product advancement, maintaining an active research department looking into all aspects of automating shoe manufacture. The task of automatically stitching and sorting shoe components has been under investigation since 1978, producing a machine called Autostitcher which is now being marketed.

The Autostitcher system uses machine vision to identify components and then stitch onto them a pattern previously defined by an operative.

Advancing technology in the form of transputers, digital signal processors (DSP's) and other fast processors has opened up the possibilities of developing the roughing, lasting and clicking machines discussed in section 1.6. In keeping with the idea of a single interconnected production line these machines will all be designed on the basis of standard hardware and processor units currently under development by BUSM.

The use of fast processors to handle data as opposed to hardware logic systems gives greater flexibility and will allow the units to be used for a greater variety of tasks. The development of these faster, more powerful units gives the opportunity for the Autostitcher design to be reviewed and for new methods to be researched with a view to improved performance.

2.2 Autostitcher - A practical example of machine vision usage

The following sections describe the mechanical and hardware aspects of BUSM's Autostitcher and also the developments underway at BUSM. The Autostitcher is based on research at Durham University by N.R. Tout into a machine for stitch marking [TOUT 1989]. Tout's research was based on initial work into the recognition of shoe components by C. Koulopoulos at City University, London [KOULOPOULOS 1982]. Figure 2.1 illustrates the main elements of the mechanical arrangement.

2.2.1 Component handling

The components are placed into the work space area by hand and taken into the machine by two pairs of gritted rollers which can grip the leather without damaging it. The rollers move the component along the y-axis while the rollers themselves are moved to give x-axis movement. In both cases motion is provided by indexed servo motors. In between the pairs of rollers is an air gap. This allows for uninterrupted scanning of the component and also for the sewing machine needle to pass through the material. For the shape to be scanned it must be smoothly taken across the air gap while the line scan camera is scanning. Although the leather components are not rigid they are held sufficiently taut between the rollers to ensure a consistent image.

The sewing machine is in a fixed position out of the camera's line of scan. Once the component has been identified and its position assessed the rollers are moved side ways to bring the component underneath the sewing machine. The actual pattern is stitched by a combination of rotating the rollers to move the component along the y axis and shifting the rollers sideways to move the component along the x axis.

The mechanical system for handling the leather was developed at Hull University in conjunction with BUSM. A more detailed discussion of the techniques involved can be found in Smith's work on the manipulation of leather workpieces [SMITH 1991].

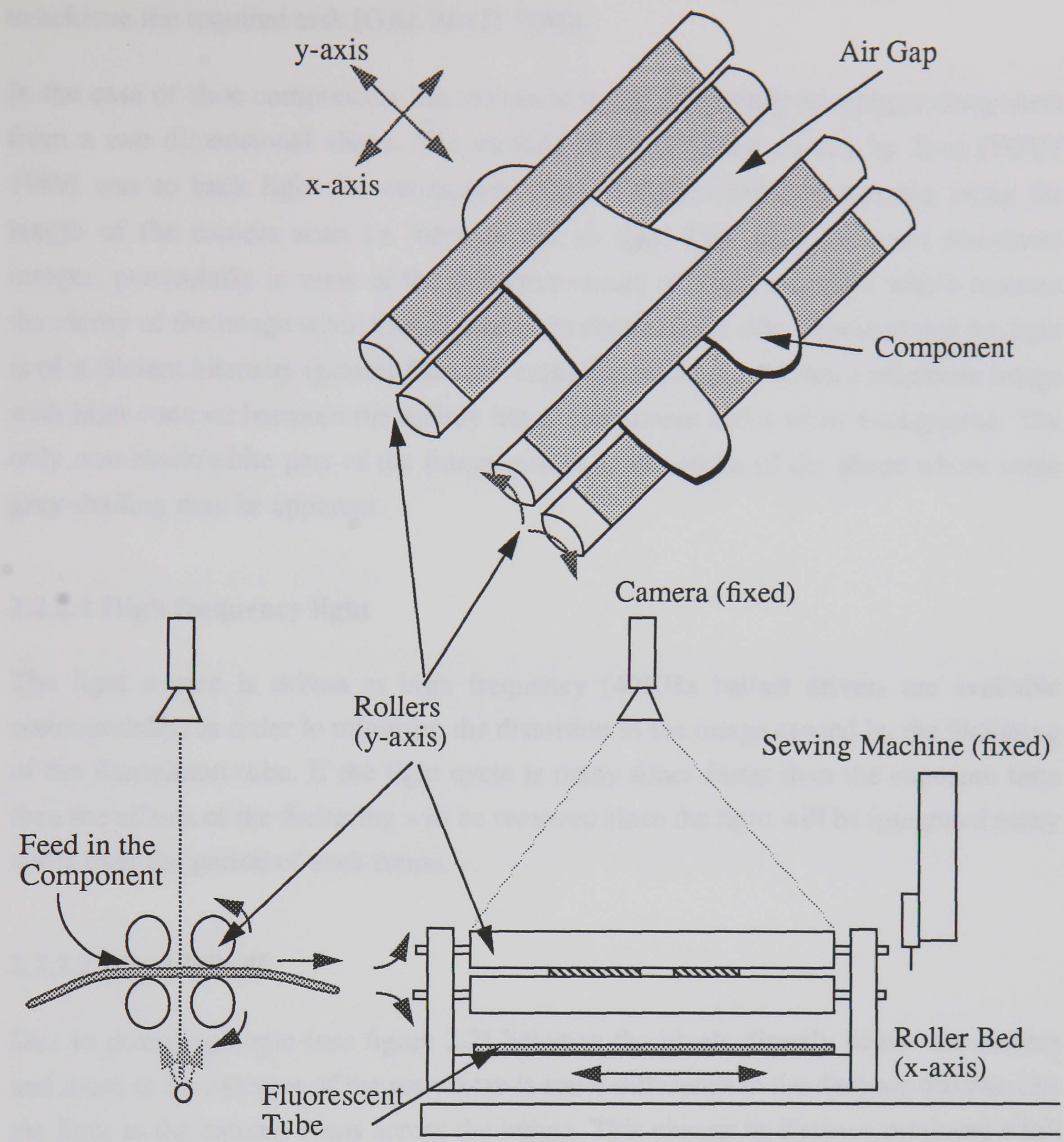


Figure 2.1 Component handling system.

2.2.2 Illumination

For any machine vision system the object(s) being viewed must be suitably lit. It is extremely unlikely that the ambient light within a working factory will be of sufficient brightness or stability to provide an adequate image and an additional light source or sources will have to be provided. The positioning of any light will depend on the object and the reason for gathering its image; the light could be positioned above the object, behind it, to the side or in any available position. Correct positioning of such lights can

enhance the quality of the obtained image and reduce the amount of processing needed to achieve the required task [GALBIATI 1990].

In the case of shoe components the system is trying to identify an opaque component from a two dimensional shape. The method of illumination chosen by Tout [TOUT 1989] was to back light the component using a fluorescent tube running along the length of the camera scan i.e. beneath the air gap. This gave the most consistent images, particularly in view of the reflective nature of some materials which reduced the clarity of the image when illuminated from above or the side. Provided that the light is of sufficient intensity (greater than 500 Lux) this method produces a silhouette image with high contrast between the mainly black component and a white background. The only non-black/white part of the image occurs at the edges of the shape where some grey shading may be apparent.

2.2.2.1 High frequency light

The light source is driven at high frequency (40KHz ballast drivers are available commercially) in order to minimise the distortion in the image caused by the flickering of the fluorescent tube. If the light cycle is many times faster than the exposure time then the effects of the flickering will be removed since the light will be integrated many times over the period of each frame.

2.2.2.2 Light fall off

Due to changing angle (see figure 2.2) between the pixels directly below the camera and those at the extreme of the scan there is some difference in the distance travelled by the light as the camera scans across the image. This change in distance produces a fall off in light levels proportional to $\cos^4\theta$ (for a discussion of this law and the associated Lambert's law see, for example, [BORN 1970 *et al.*]). The maximum angle change in Autostitcher is 0.2 radians so the light received from the extreme edges of the scan is 92.3% of that directly underneath the camera. The camera thresholds the signal at 50% so the effect of this light fall off is insignificant for a binary system.

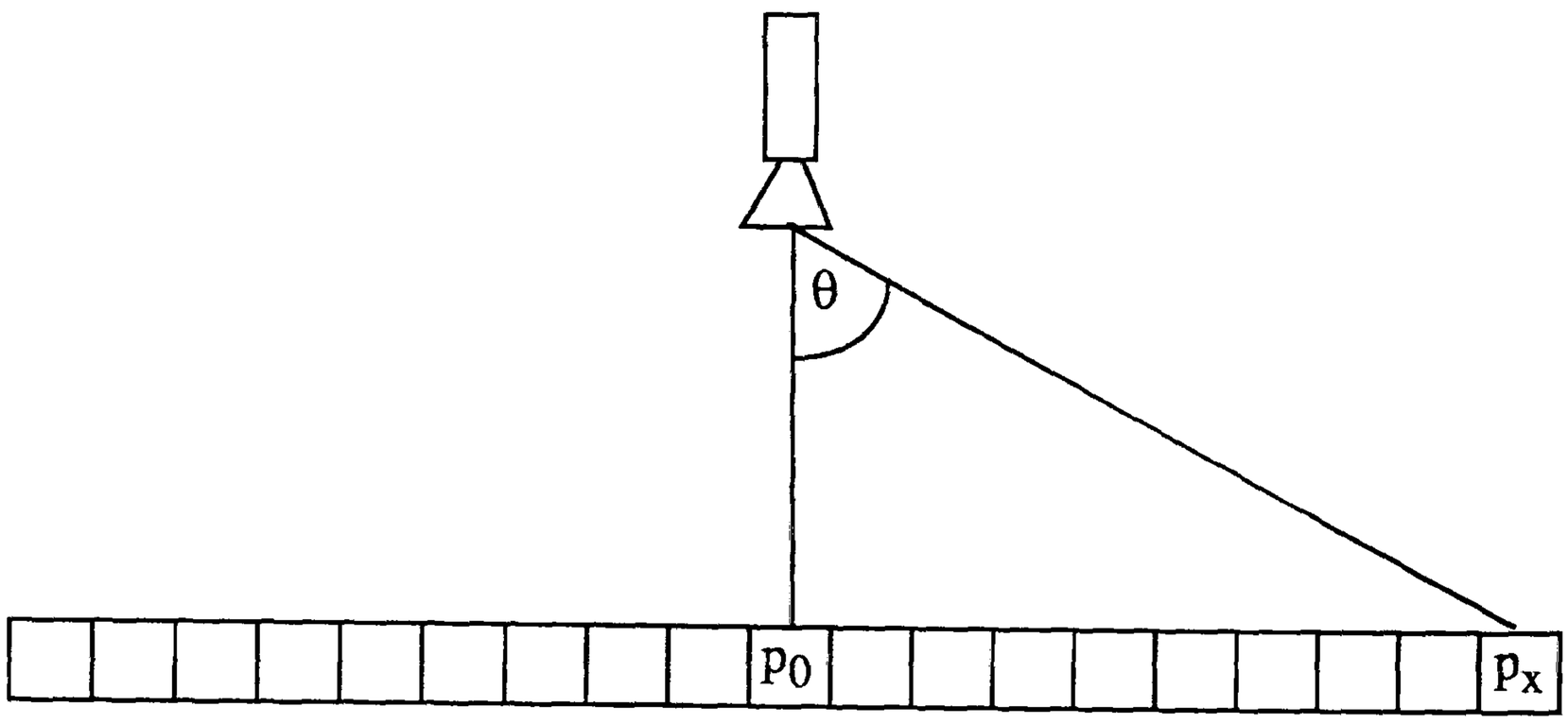


Figure 2.2 Relation between image pixels and camera.

2.2.3 Camera and optics

The choice of camera is almost unlimited and will depend on the required function of the system. The three main categories available are:-

- A line scan camera which captures the scene one line at a time.
- An area scan camera which typically captures several hundred lines at a time.
- A Laser scanner.
- Additionally cameras may be colour or monochrome/grey scale and come in a variety of pixel array sizes: 256 by 256, 512 by 512, 1024 by 1024, 2048 by 2048 or 4096 by 4096 for an area scan camera and 256, 512, 1024, 2048 or 4096 for linescan cameras.

Since any colour in the viewed scene is of no interest only a grey scale camera need be chosen. Linescan cameras are less expensive than their area scan or laser counterparts for a given resolution and have the advantage over area scan cameras of requiring no additional processing to scan components of excessive length (e.g. boots) which might require several related frames of an area scan image.

2.2.3.1 Aspect ratio

The ratio between the x axis pixel spacing and the y axis pixel spacing is known as the aspect ratio. In area scan cameras it is usually fixed by the manufacturers so that the x pixels are spaced 1.33 time the distance between the y pixels. This ratio is used due to

the standard devised for entertainment television cameras. With a line scan system only the x pixel spacing is set by the geometry of the camera equipment, the y pixel spacing will depend on the speed of the component beneath the camera. Thus the aspect ratio can be set to unity to give a more even image.

2.2.3.2 Component movement

The disadvantage of linescan cameras is that components must be moved at a known speed in a direction perpendicular to the scanning array. They are useful in areas involving inspection of moving conveyors where the component is already on the move. If components are to be stitched then they must be moved by the system thus the movement required for a linescan camera is an integral part of the system.

2.2.3.3 Grey scale images

Grey scale images typically describe each pixel within an image using an 8 bit word resulting in a value between 0 and 255. This has obvious consequences in terms of storage costs but the grey scale representation can be used to interpolate between the actual pixels and approximate an image that might be produced by a binary camera of higher resolution.

2.2.3.4 Binary images

This is the simplest form of image requiring only 1 bit to describe each pixel thus reducing storage costs and data transmission rates. For the purpose of identifying shoe components a binary image is sufficient provided a camera of high enough resolution is used.

In practice non colour cameras give out a single analogue signal between 0 Volts and 1 Volt. This can then be transformed into a binary (1 bit) data stream, a standard grey level (8 bit) data stream or any other level of digitised bits by the use of analogue to digital converters.

2.2.3.5 Camera choice

The type of camera actually selected by Tout [TOUT, 1989] for the purpose of capturing shoe component images was a monochrome, line scan camera with 2048 elements. The analogue output from the camera is thresholded at 50% to generate a binary signal which is sent to the hardware. The choice of array size was based on a

combination of the best commercial value and the size of the likely components which was expected to be between 2.5 cm^2 to 25 cm by 30 cm.

A 50mm, f5.6 lens is used to focus the image onto the camera array and is mounted at a height such that the object plane pixels are spaced at 0.2 mm. The increments of the component are also set to 0.2mm per scan thus the resolution of the system is 0.2mm by 0.2mm, giving a nominally unity aspect ratio and a maximum component width of 40.96cm.

The camera array is non shuttered and continually exposed to the object plane. Under the control of a master clock the camera is given a sync pulse every 2048 cycles causing the charge in the CCD elements to be transferred in parallel to corresponding shift register cells. During the following 2048 clock cycles each cell is transferred serially out of the camera to the hardware. In sequence with the camera sync pulse a signal is sent to the motor controller instigating a 0.2mm advance of the component by rotation of the rollers.

Appendix II gives details of the operation of a linescan camera.

2.2.3.6 Material thickness

Tout discussed the effect of the thickness of material on the image and considered that although the error was small it could potentially alter the recognition performance. Figure 2.3 shows how the actual edge of a component differs from the apparent edge as seen by the camera. The magnitude of the error (e) for each edge (a scan line must have at least two edges) will depend on component thickness (t), its distance (w) from the axis perpendicular to the camera and the height of the camera (d) as shown by equation 2.1.

A shoe component 2.5mm thick and 200mm wide centrally positioned 1100mm beneath the camera will have an apparent width of 200.5mm. If more than two edges are contained within a line then the error will be worse. BUSM's specification (chapter 3) sets out that any point on the component is to be located with an accuracy of $\pm$ 0.25 mm. The material thickness could, therefore, affect the accuracy of component location as well as identity.

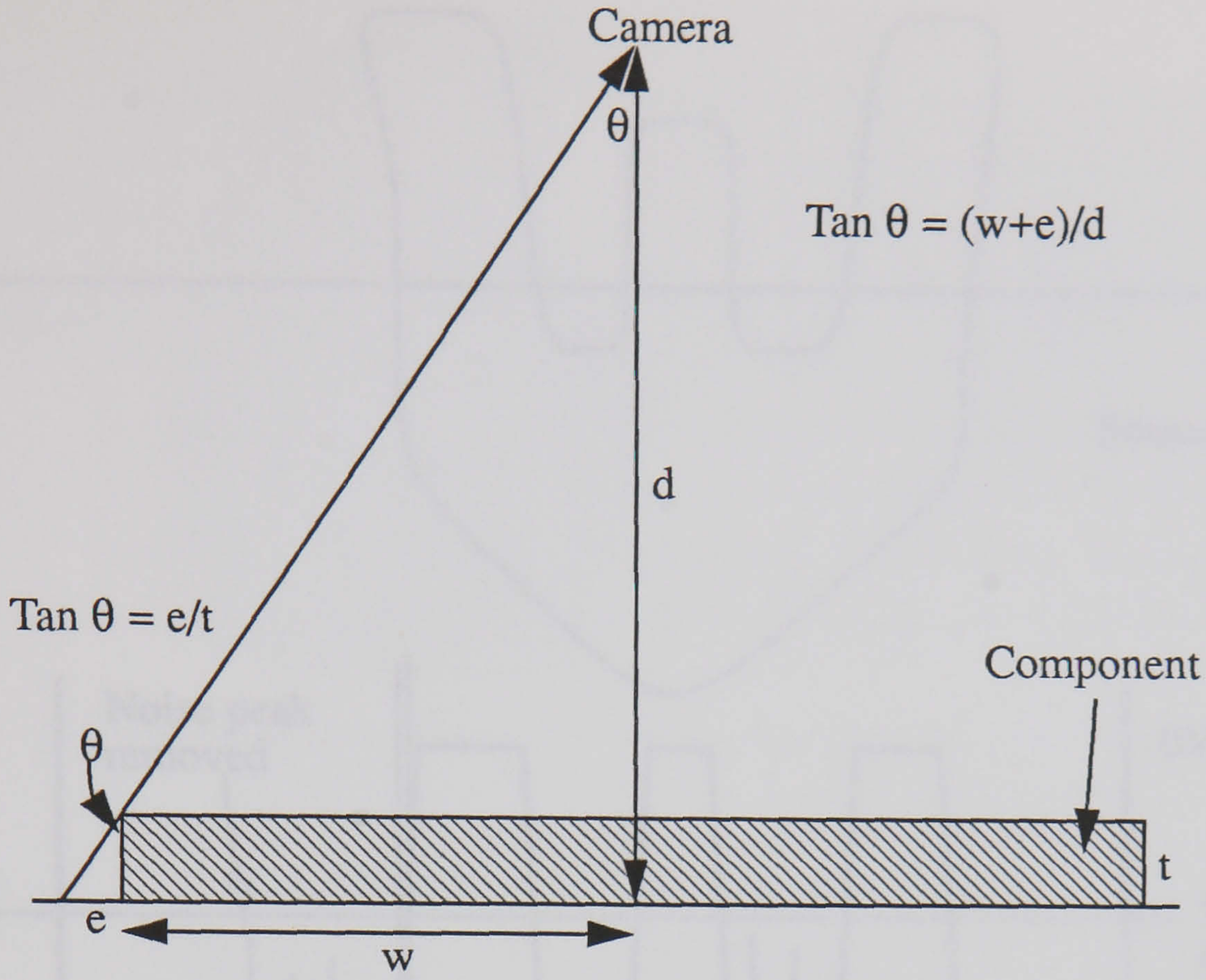


Figure 2.3 Error due to material thickness.

$$e = \frac{tw}{(d-t)} \quad \text{.....eq 2.1}$$

2.2.4 Threshold selection

The use of analogue to digital conversion requires the setting of thresholds to specify each digital level. In the case of a binary signal only one threshold is required. The value of the threshold lies between 0 Volts and 1 Volt and can be set so as to minimise noise problems at a later stage in the image processing functions of the system. Leather fibres are one such source of noise. A typical leather fibre is less than the width of a pixel at the image plane and results in an analogue value between 0 volts and 1 volt. Application of a suitable threshold can remove the fibres from the image without the use of processing power.

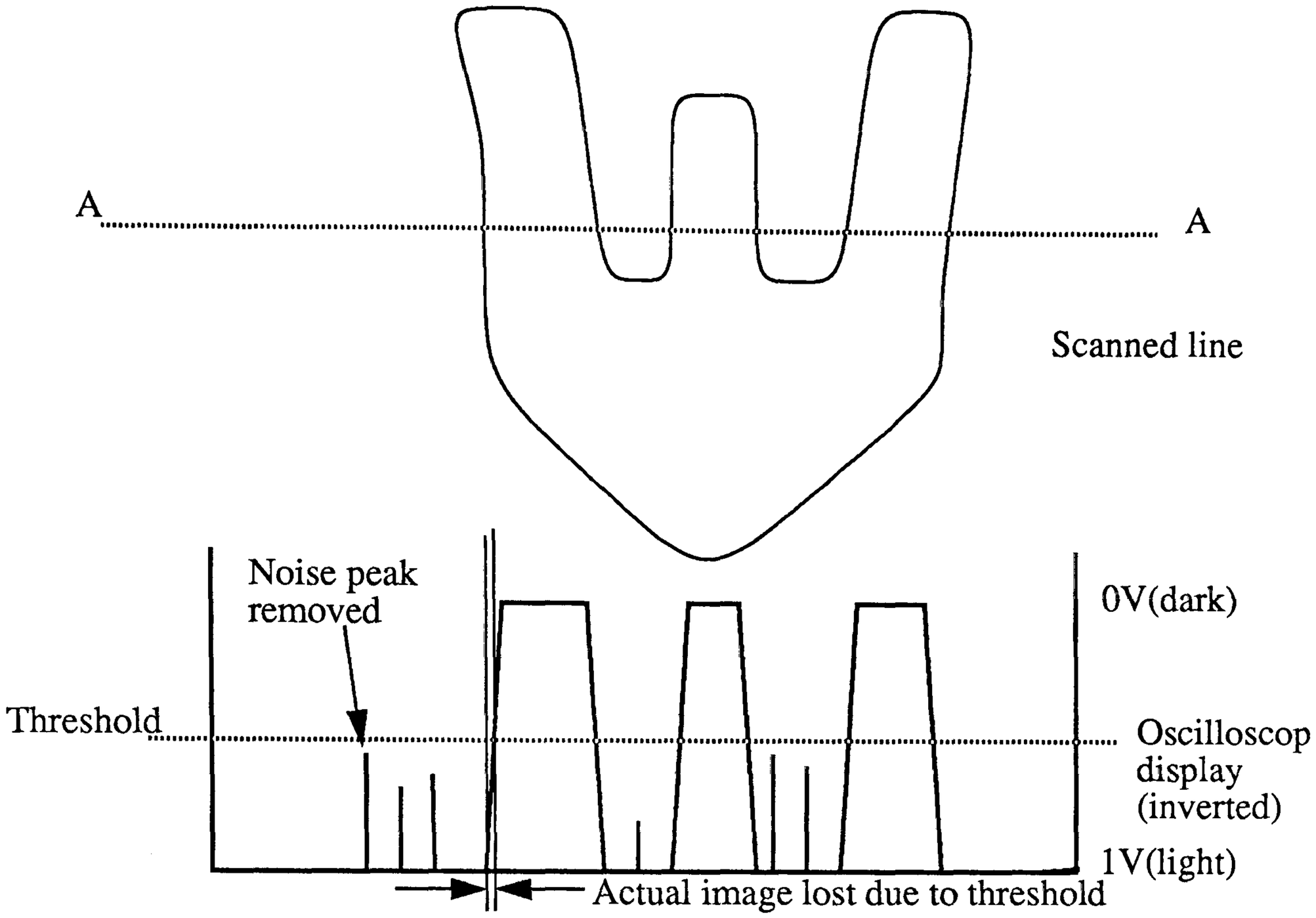


Figure 2.4 Effects of threshold.

Figure 2.4 shows a typical output on an oscilloscope as produced by a linescan camera scanning across section AA. A camera pixel saturated with light will contain charge equivalent to 1V. Those pixels effectively covered by the observed object will collect no charge, equivalent to 0V. A semi covered pixel will collect a proportion of charge producing a signal in the range 0V to 1V. By setting a threshold above the level of these intermediate signals the noise due to loose fibres can be removed. Tout [TOUT 1989] selected a threshold level of 50% or 0.5V.

Fig 2.5 shows a section of an image (including an unwanted spot) and the corresponding camera output for that section of one line.

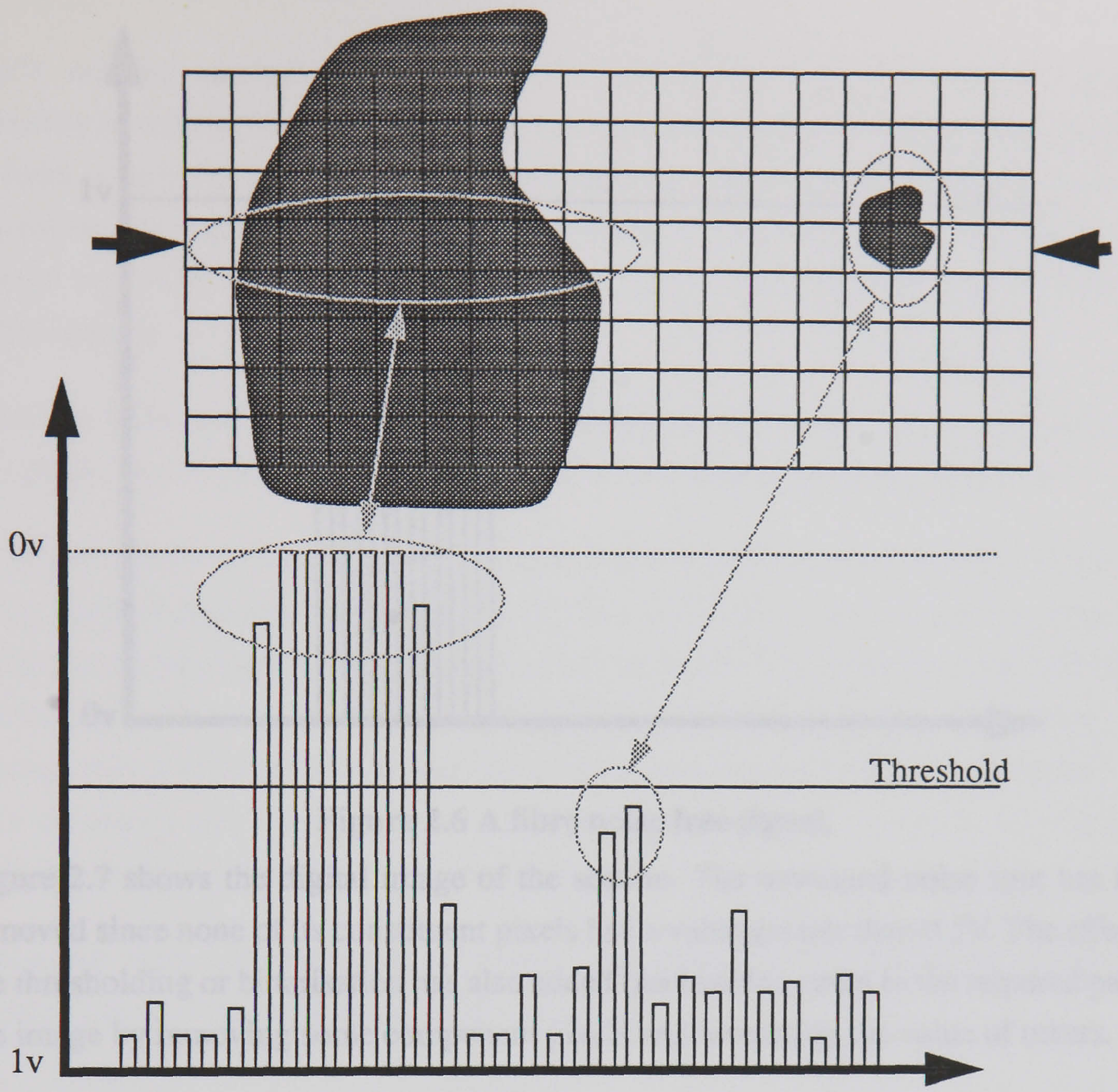


Figure 2.5 Removal of fibre noise.

When the threshold of 50% is applied to the analogue signal any pixels with a value less than 0.5V are set to 0V and the remainder to 1V. The corresponding section of this new digital signal is shown in figure 2.6.

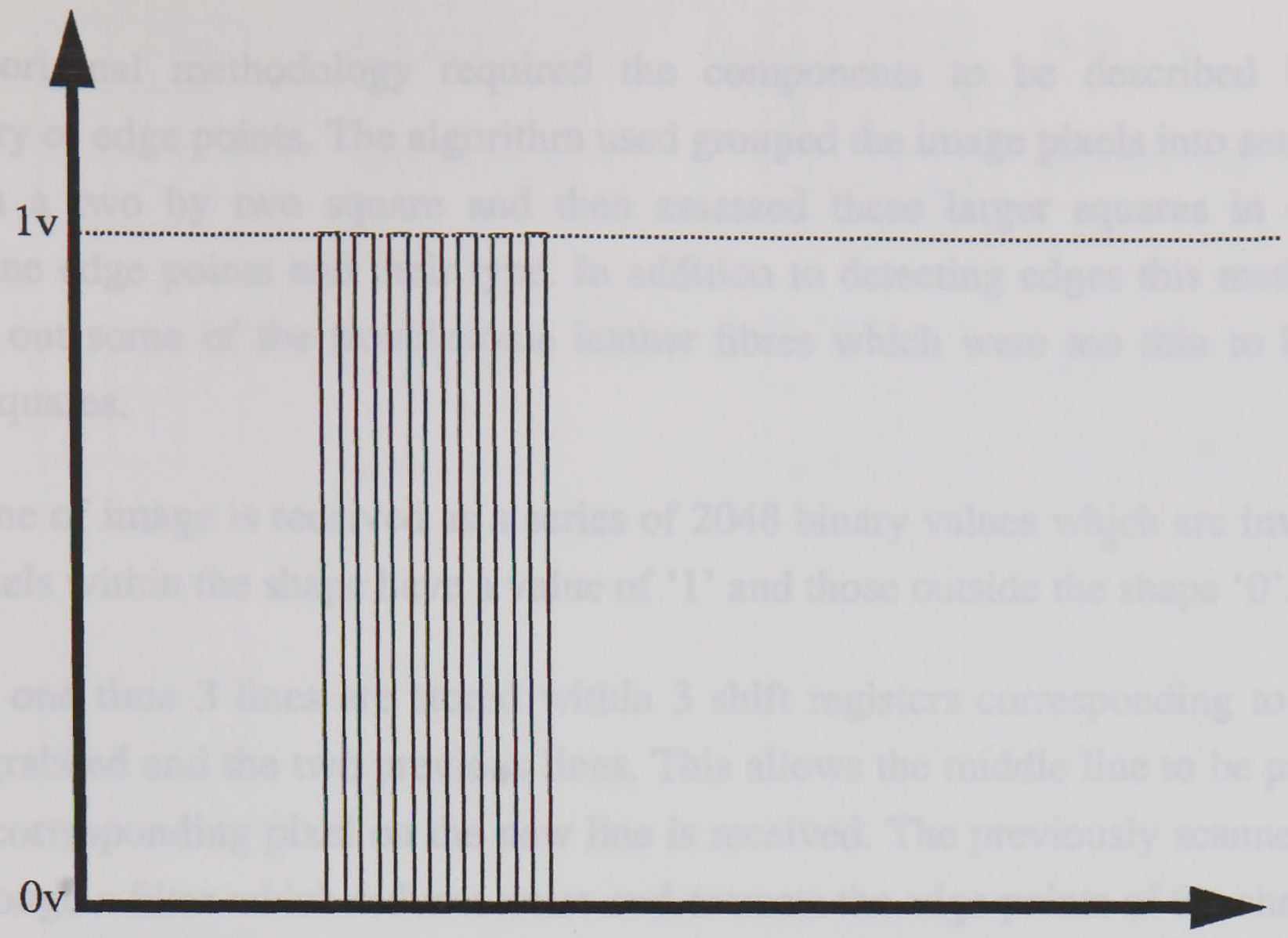


Figure 2.6 A fibre noise free signal.

Figure 2.7 shows the digital image of the section. The unwanted noise spot has been removed since none of its constituent pixels had a value greater than 0.5V. The effect of the thresholding or binarisation has also added quantisation noise to the required part of the image by removing some component pixels and increasing the value of others.

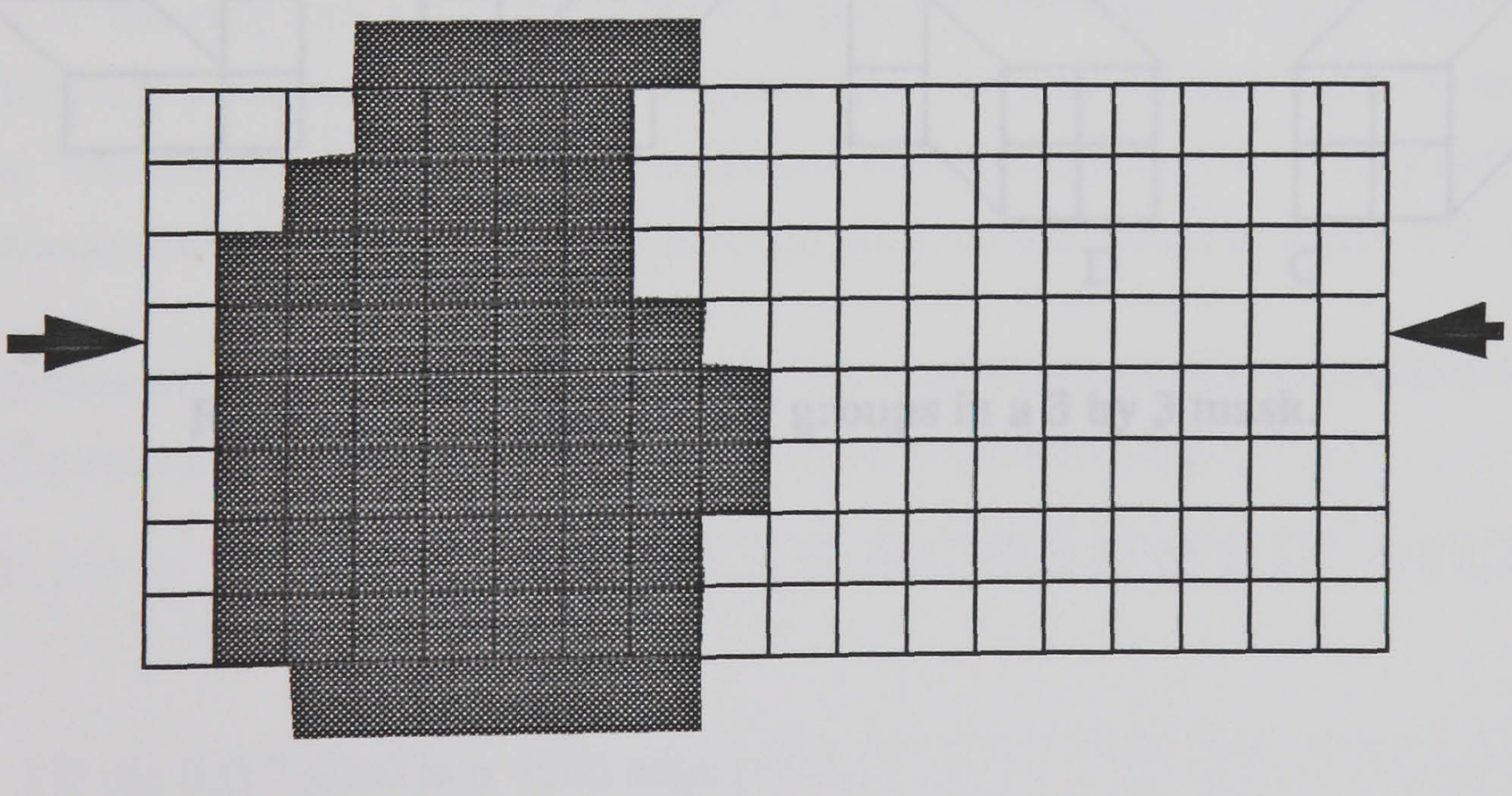


Figure 2.7 The fibre noise free image.

2.2.5 Filtering and edge detection

Tout's original methodology required the components to be described by their boundary or edge points. The algorithm used grouped the image pixels into sets of four to form a two by two square and then assessed these larger squares in order to determine edge points and their type. In addition to detecting edges this method also filtered out some of the troublesome leather fibres which were too thin to form the larger squares.

Each line of image is received as a series of 2048 binary values which are inverted so that pixels within the shape have a value of '1' and those outside the shape '0'.

At any one time 3 lines are stored within 3 shift registers corresponding to the line being grabbed and the two previous lines. This allows the middle line to be processed as the corresponding pixel on the new line is received. The previously scanned line is run through a filter which reduces noise and extracts the edge points of the shape. This process works on-line in hardware but always lags one scan behind the camera. The type of edge point along with its x and y co-ordinates are stored in a set of shift registers ready to be read in by the software.

There are eight possible types of edges characterised by combinations of the four corner groups of pixels in the three by three mask.

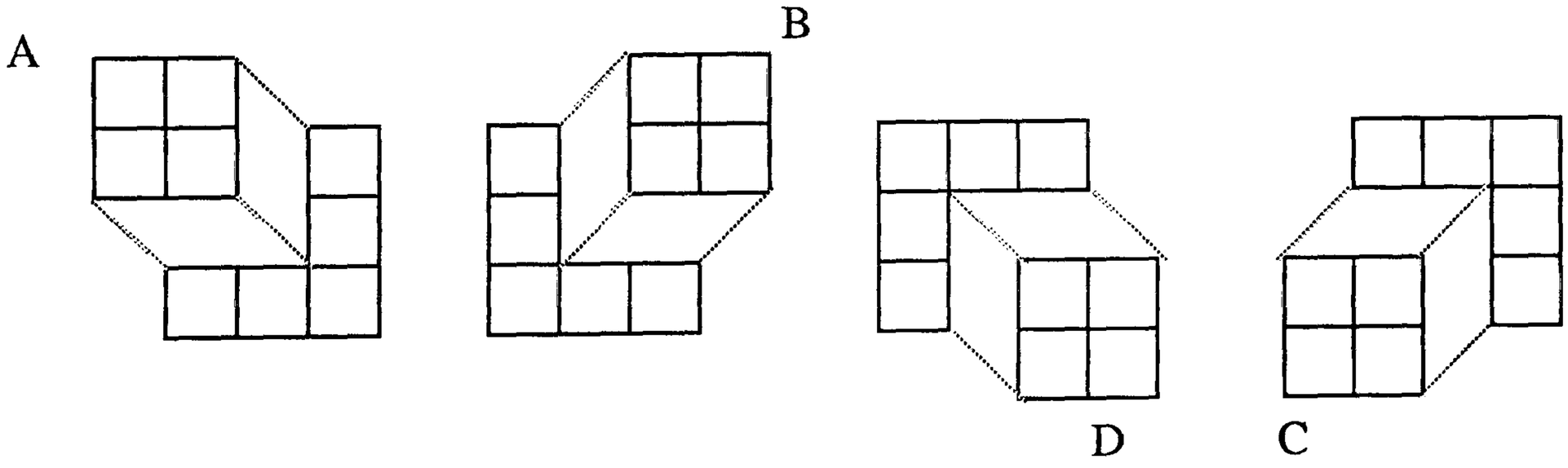


Figure 2.8 The four corner groups in a 3 by 3 mask.

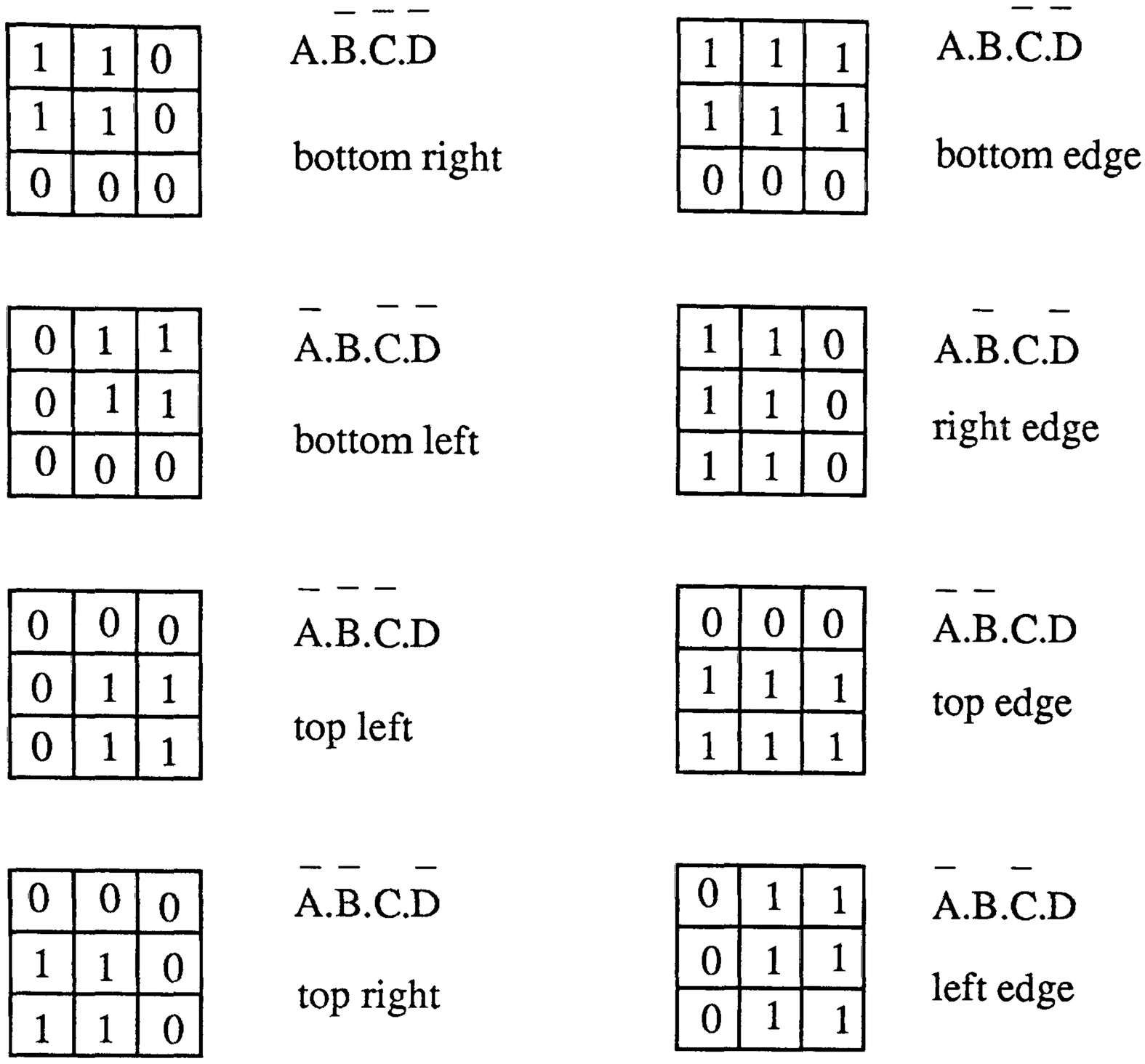


Figure 2.9 The eight edge types.

The filtering operation is performed for each pixel by considering it and its eight neighbours by first evaluating the corner groups A, B, C and D as shown in figure 2.8. Each corner group is set high if all the respective pixels are set high but is set low if one or more of the respective pixels are set low (i.e. by ANDing the pixel values). This means that any pixel not contained in a square group of four or more pixels is removed as noise. The corner groups are then combined and evaluated to see if they satisfy the requirements for the centre pixel to be an edge point as illustrated in figure 2.9.

The edge detection filter can be described by the combinations of corner groups which are valid edges as shown in figure 2.9 and shown by equation 2.2.

$$\mathrm{P(i,j)} = A \oplus B \oplus C \oplus D \oplus \mathrm{A.B} \oplus \mathrm{A.C} \oplus \mathrm{B.D} \oplus \mathrm{C.D} \ldots eq2.2$$

P(i,j) = 1 if the $(i,j)^{th}$ pixel in a valid edge pixel;

P(i,j) = 0 otherwise

This expression and the evaluation of the corner groups is evaluated using the hardware arrangement shown in figure 2.10

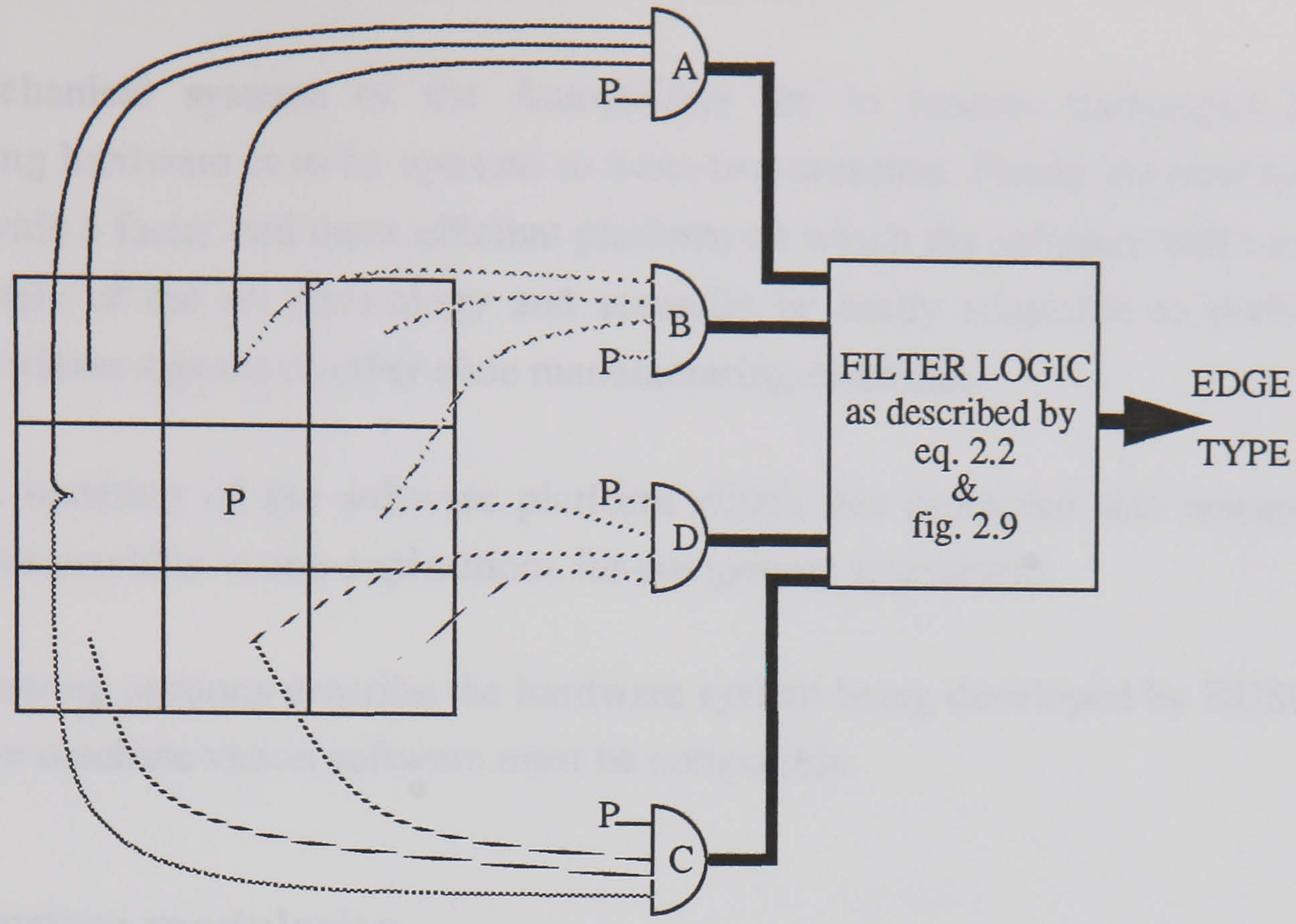


Figure 2.10 Hardware arrangement for edge detection.

2.2.6 Software

The component image is completely described by the edge representation which the hardware has extracted and can be read in by the software. The hardware has compressed the data description of the shape and thus reduced the data transfer time as compared with that for a complete image. The software can now use this information to identify the component and locate its position. The methods used to do this are investigated in chapters 3, 6, 7 and 8. The identity and position data are then used (in the case of a stitching machine) to stitch the pattern onto the component.

2.2.7 Stitching the component

The fixed sewing machine is controlled directly from the processor with the component moved beneath it by rotation of the rollers for y axis travel and movement of the roller unit for x axis travel.

Finally the component is ejected from the system which then awaits the next component.

2.3 BUSM's proposed hardware developments

The mechanical systems of the Autostitcher are to remain unchanged but the controlling hardware is to be updated to meet two criterion. Firstly the new hardware will provide a faster and more efficient platform on which the software will run by the use of state of the art technology and secondly be easily adaptable to perform the machine vision aspects of other shoe manufacturing machines.

It is the updating of the software platform which has prompted this research into alternative machine vision applications for component assessment.

The following sections describe the hardware system being developed by BUSM with which the machine vision software must be compatible.

2.3.1 System modularity

Each system will be constructed from a set of modules with each module used for a different task. In the case of the stitching machine the required modules will be;

(a) Vision unit

(b) Motor controller

(c) Disk unit

(d) Additional transputers if required

(e) Sewing machine unit.

All units are mounted on a mother board controlled using transputers. The hardware in each unit will be essentially the same. Differing tasks will be achieved by using differing software. This thesis is concerned with researching processes to work on the vision unit module.

2.3.2 The machine vision front end or module

This module includes the camera interface and all the necessary hardware to process line scan images. Processing power is provided by a Digital Signal Processor (DSP) which deals with the raw camera data. This is followed by a series of transputers to

analyse the image and produce a set of values detailing identity, orientation and position.

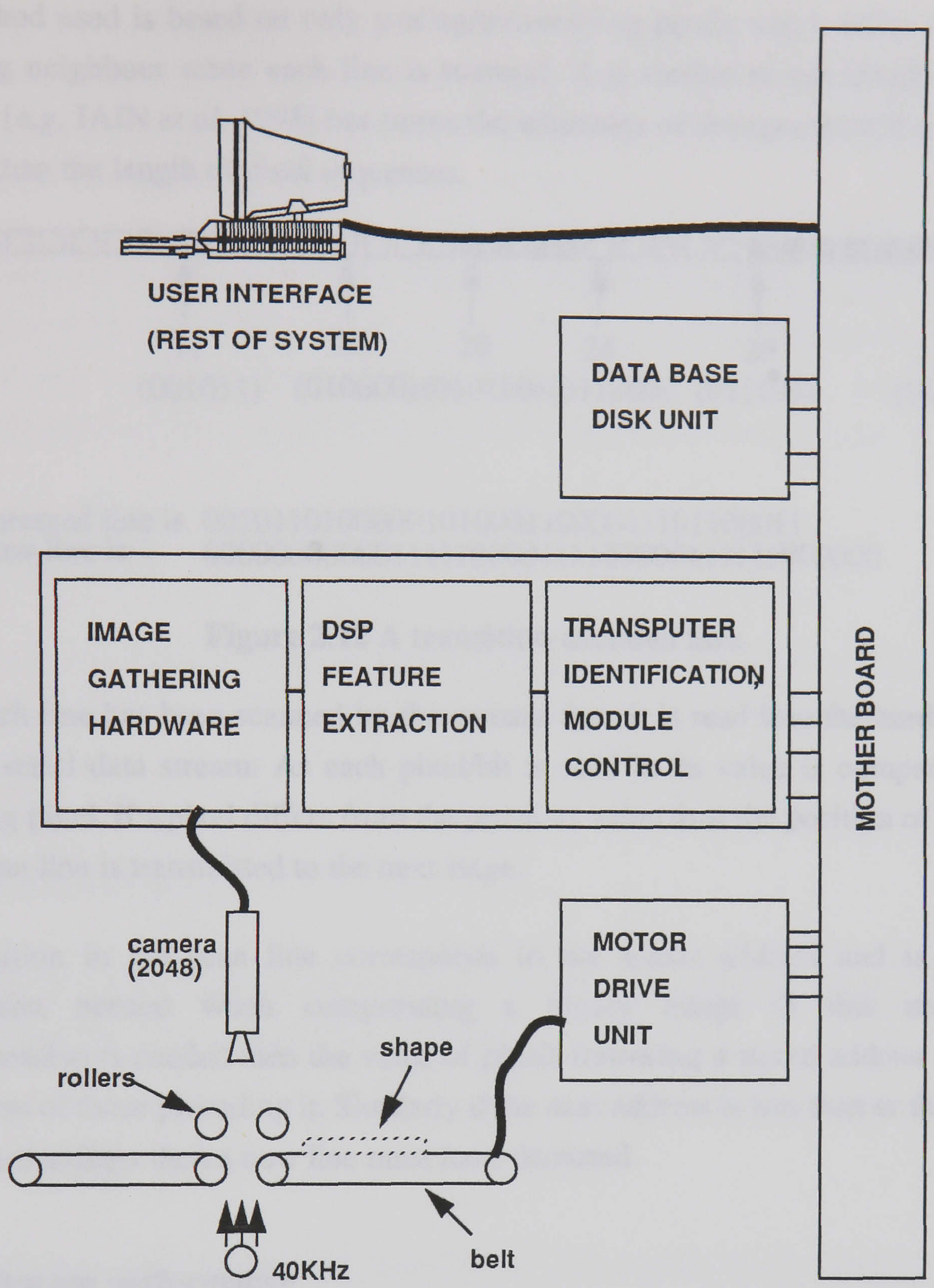


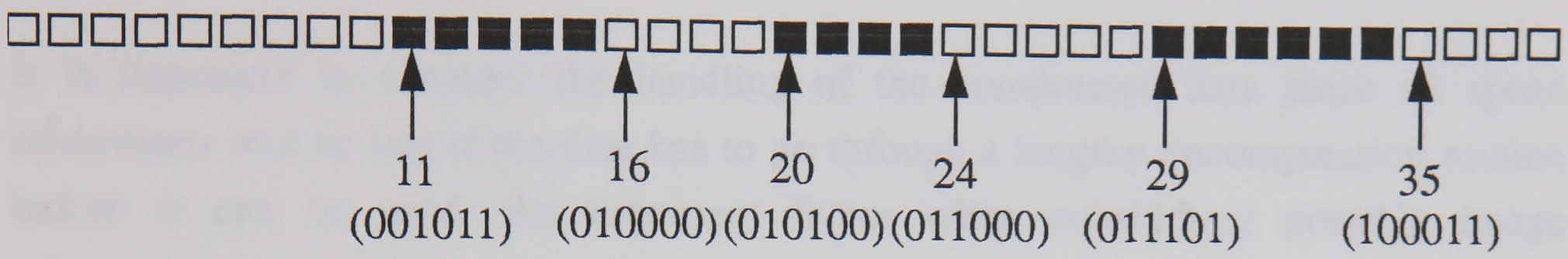
Figure 2.11 A vision unit.

2.3.3 Image compression

It is important to be able to minimise the representation of an image since this reduces the storage requirement and the amount of data transmitted around the system. A binary image 2048 pixels by 2048 pixels requires in excess of 4 Million bits if it is handled in its raw state. Relatively simple compression techniques can be used to reduce the bit requirement of an image.

2.3.3.1 Transition encoding

The method used is based on only storing/transmitting pixels which differ from their preceding neighbour when each line is scanned. It is similar to run length encoding methods [e.g. JAIN *et al.* 1995] but stores the addresses of changes from 0 to 1 or 1 to 0 rather than the length of pixel sequences.



Compressed line is 001011010000010100011000011101100011
Raw line is 00000000000111110000111100000111111000000

Figure 2.12 A transition encoded line.

Once each line has been scanned by the camera then it is read into the hardware as a 2048bit serial data stream. As each pixel/bit is read in its value is compared to the preceding pixel. If a pixel differs from the previous value then the position of that pixel in the scan line is transmitted to the next stage.

The position in the scan line corresponds to the x-axis address and is the only information needed when compressing a binary image in this manner. If uncompression is needed then the value of pixels following a stored address is simply the inverse of those preceding it. Similarly if the next address is less than or the same as the current address then a new line must have occurred.

2.3.3.2 Storage performance

The width of 1 scan is 2048 pixels numbered from 0 to 2047. It requires 11 bits to describe any address in this range. There are on average four changes per line which require 44 bits in total to describe their addresses. This compares with the 2048 bits required to describe the binary state of every pixel individually. The average rate of compression per line is thus 44 / 2048 or 97.85%

2.3.4 Speed performance

Despite the use of fast transputer and DSP technology it is the software element of the system which determines the speed of the overall process. There are two main areas

which must be addressed in respect of the software; the acquisition of data and the processing of that data. It has been found that the transputers while being fast in data processing are considerably slower in accessing outside data sources. Thus the compression of data for hardware/software transmission at the earliest opportunity is an important consideration. The compression algorithm used reduces the data by 97.85% this increases the rate of image transmission 46 times.

It is important to consider the handling of the compressed data since all speed advantages will be lost if the data has to go through a lengthy uncompression routine before it can be used. An important factor when considering possible image representations was their compatibility with the compressed image format.

2.3.5 Processors

The compressed image is read in by the digital signal processor one line at a time. The DSP then calculates values for the image and passes them onto the transputer for image analysis.

2.3.5.1 Synchronisation

Synchronisation between the hardware, the DSP and the transputer is avoided by the use of buffers for the passing of data and code

2.3.6 Filtering

BUSM intend to replace the edge detection and filtering hardware used by Tout with software running on the DSP. Thus, the hardware contains no dedicated filtering other than that provided by the thresholding operation. This arrangement is to allow for greater flexibility within the vision module.

Appendix III contains a more detailed explanation of the system communications.

2.4 Summary

This chapter has described the mechanical and hardware systems for which this research has sought to identify improved processes of identification and position assessment.

CHAPTER 3

REVIEW OF THE PROBLEM AND PREVIOUS WORK

3.1 The problem

At the earliest concept stage of the project in 1979 [CROUCH 1979] BUSM set out the target specification for the performance of a machine required to identify shoe upper components and evaluate their orientation and position. This called for the following:

- Component location to within ±0.25 mm.
- Orientation measured to an accuracy of ±0.1°.
- A database size of 20,000 components.
- A rejection rate of less than 1 in 1,000 (0.1%).
- A substitution rate of less than 1 in 10,000 (0.01%).
- The total process time to be less than 5 seconds per component.

Rejection is the failure to assign a component to any class and substitution is assigning a component to the wrong class. Rejection is the preferred failure since substitution can lead to a wasted component and/or damaged machinery.

Shoe components are generated randomly by the Clicker in order to make best use of the material available. This means that the next process in the production line receives a random steam of components both in terms of identity and position. This subsequent process is typically one of the following four;

(a) Sort the components into kits corresponding to a pair of shoes followed by pre-stitching the components within the kit.

(b) Group components into batches of same shapes followed by pre-stitching with same pattern.

(c) Pre-stitch components individually and then sort into kits corresponding to a pair of shoes.

(d) Pre-stitch components individually and then sort into batches of identical pattern/ shape.

There are then, two types of machine which could be developed in line with these processes. The first need only identify a component so that it can automatically be sorted into the correct kit. The second, more advanced machine, needs to identify a component including its size, ascertain its position and assess its alignment. The correct pattern can then be selected, adjusted and stitched into place.

The problem of identification in this type of system is relatively large compared to most machine vision identification problems. A typical shoe factory might produce 25 designs of shoe with 20 components for each of the left and right sides giving 1,000 different shapes that might be encountered. Each shoe is produced in 20 half size variations giving some 20,000 different components that might be found in a typical database.

3.2 A general machine vision solution

There are two distinct phases of operation to consider:

(a) The training phase

The system must be 'trained' on each type of component it is expected to recognise. During this phase a camera is used to obtain a representation of a prototype shape and this image is stored either complete or as a set of representative features within a database and allocated a label corresponding to that shape. In the case of a stitching machine this data base entry would also include the co-ordinate points of a pattern to be stitched.

(b) The recognition or operational phase

A camera is once again used to obtain an image of a shape and software or hardware used to match the image or its features with those of a prototype shape stored in the database. Once a match has been found then the position and orientation of the shape can be assessed. The co-ordinate points stored in the data base can then be adjusted and

used to control devices operating on the component such as a sewing machine in the case of an automatic stitching system.

3.3 Variation between components

In simplistic terms the problem of identifying shoe parts is simply a case of comparing an unknown shape with each shape in the data base until a match is found. In practice each nominally identical shape will be different due to a variety of factors generalised as noise.

This noise can be caused by the imaging system, the production system or the nature of the component material.

3.3.1 Imaging system variations

- Varying light levels including reflectivity of materials lead to differing images.
- Image acquisition introduces quantisation noise which will vary with position and rotation.

3.3.2 Component property variations

- The flexible nature of shoe materials can produce distortions in shapes.
- Leather is anisotropic so the stretching/shrinking properties vary with direction.
- Leather shapes will vary with time due to changing humidity levels.

3.3.3 Production variations

- When leather is cut the edge tends to be distorted by loose fibres.
- The knives used to cut the leather are only manufactured nominally identical and can be further distorted through mishandling.

3.4 Previous work on shape description

In order to identify a component from an image there must be some method by which that image can be compared with those previously shown to the system. The use of the entire image is cumbersome and slow due to the excessive amount of information held

in such an image. An image of 2048 pixels by 2048 pixels contains over 4Mbytes of information for a binary scan and over 33Mbytes for an 8 bit grey scale scan. The amount of information can be reduced by extracting from an image a set of measures or features which describe the component sufficiently well for the purpose of identification.

Koulopoulos [KOULOPOULOS 1982] investigated ways of extracting features from the shape images so that they could be identified, located and orientated. His conclusion was that the problem was best solved by a hybrid system involving edge detection, radii calculation, shape centroid location and shape area. Due to the limits of the available technology his system took 5 minutes to process each shape but did demonstrate that machine vision could be applied to the shoe industry.

Tout [TOUT 1989] took the work of Koulopoulos and incorporated it into a stitch marking machine for physically marking patterns onto shapes. He expanded the feature vector to include second order moment invariants and extra radii to give improved orientation assessment but found that he could not improve on the general methods of Koulopoulos (other than improved technology for greater speed). Tout's stitchmarking machine research later formed the basis of BUSM's automatic stitching machine.

The following sections review the methods and problems considered by Koulopoulos and Tout.

3.4.1 Assumptions

Koulopoulos made the following two assumptions:

- That we are dealing with discrete objects, i.e. objects which do not overlap.
- The objects are to be described one at a time.

In addition to these it will be assumed that objects will only be presented in single file but not necessarily uniformly spaced.

3.4.2 Criteria of selected features

There are many different features which could be selected for the purpose of shape representation. Koulopoulos suggested the following desirable qualities for such measures:

- Uniqueness.
- Independence between features used in each set.
- Rotation invariance.
- Scale invariance.
- Straightforward physical interpretation.
- Parsimony. (The number of features used is kept to a minimum).

The use of features having only a straightforward physical interpretation severely limits the choice while scale invariant features means that identical shapes in differing sizes cannot be distinguished.

3.4.3 Complications in shapes

Several aspects of a component may complicate its identification. Many components have the same outline but differ due to the inclusion of 'holes' within the boundary. Some components have re-entrant boundaries such that a radius from the centre of the component crosses the boundary more than once.

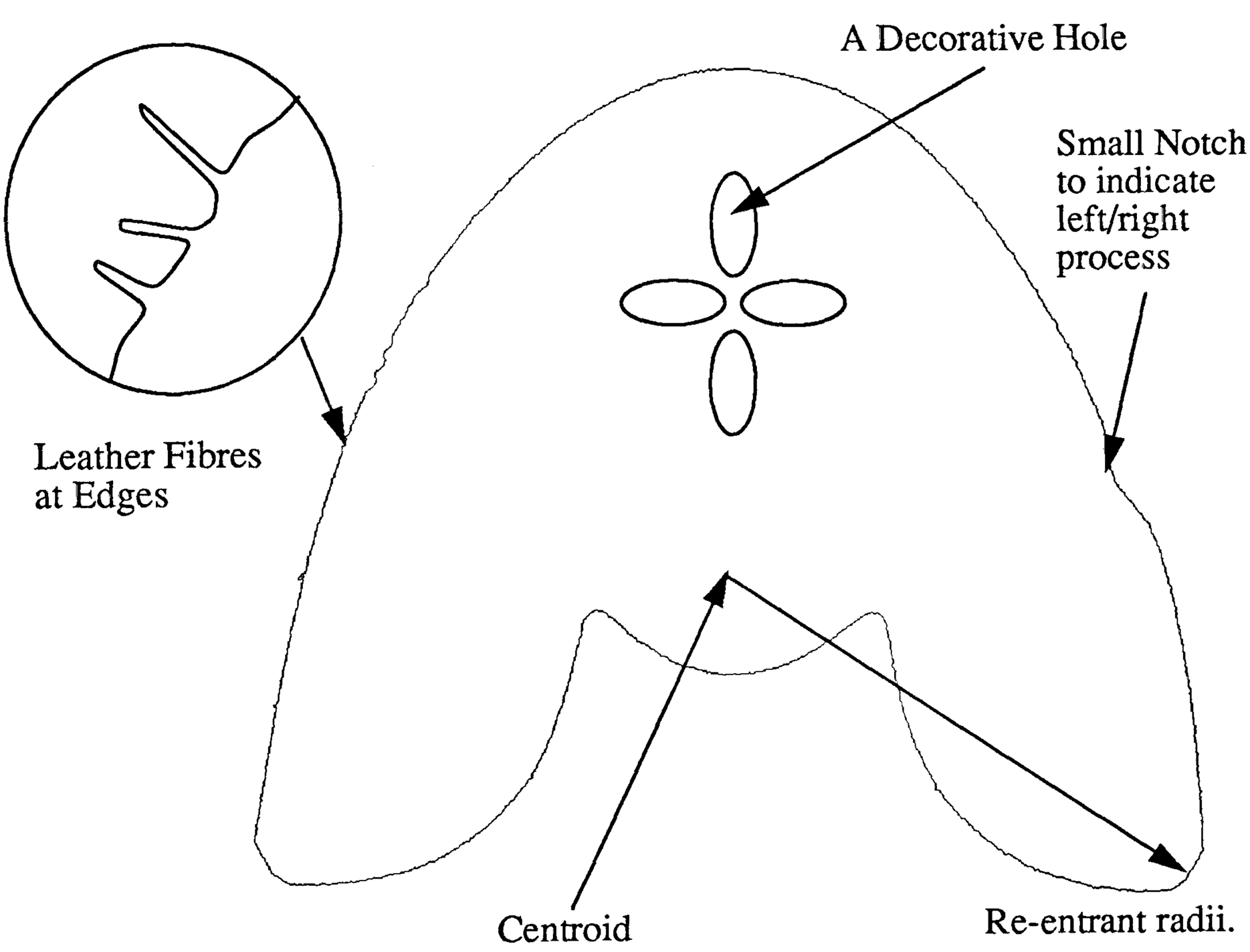


Figure 3.1 A typical shoe upper component.

3.4.4 Features considered

The following sections review the shape representation methods considered by Koulopoulos.

3.4.4.1 Heuristic measures

These are perhaps the most simple features that can be extracted and include measures such as area, perimeter length and the shape factor defined as the square of the perimeter divided by the area. Koulopoulos rejected all methods based on the length of shape perimeters because of the difficulty in getting accurate measurements. The area feature was more robust and was retained.

The perimeter pixels of a shape are the most noisy due to image quantisation and measures such as perimeter length can vary considerable between examples of identical shapes. In contrast with this the area of a shape involves every pixel within the shape boundary and is thus resilient to noise. Features involving every pixel in this way are known as a global feature.

3.4.4.2 Fourier methods

Two methods are considered here.

The first involves plotting a series of radii from centroid to boundary at regular intervals and has been used to inspect quartz grains [BOON 1982 *et al.*] and biological cell structures [WALI 1991 *et al.*]. These radii can be considered as the amplitudes of a 1 dimensional signal from which Fourier coefficients (and phase) can be extracted. The coefficients are rotationally invariant and can be compared to other sets of coefficients in order to identify the shape. Any holes or re-entrant boundaries distort the radii, making the system unsuitable for shoe component recognition.

The bend function method [ZAHN *et al.* 1974] can handle re-entrant shapes but not holes. The boundary is followed, sampled at fixed lengths and the angle moved through is recorded. The recorded angles can then be used to form a 1 dimensional signal as above.

3.4.4.3 Density function methods

A density function representation of a shape describes it by the occurrences of certain elements of the image. Koulopoulos considered four such methods:

The Slope density of a component as used by Nahin [NAHIN 1972] is the proportion of lines in the outline lying at each given angle. Rotating the shape alters the slope density, thus if components were to be identified then several sets of density measurements would be needed for each shape to cover differing orientations.

Nahin's method requires the components outline to consist of relatively long straight sections in order that their alignment can be calculated. Freeman's directionality spectrum [FREEMAN 1962] uses the occurrences of each element of the chain code (see 5.1.1.1) in order to produce a directionality spectrum. Due to the nature of the chain code only eight angles can be evaluated. The directionality spectrum can, however, be used for any continuous outline no matter how frequent the changes of direction may be. Rotation of the shape will also alter the directionality spectrum.

Brezina's radius method [BREZINA 1977] simply obtains the distribution of the radii from the centroid of the shape to the edge. In order to achieve this the centroid of the shape must first be calculated. The method does, however, have the advantage of being rotationally invariant.

The final density method considered by Koulopoulos was that suggested by Piper [PIPER 1970]. A chord of fixed length is moved around the outline of the component. The angles between each successive chord are then used to form a cumulative frequency curve. The distribution of the angles between each pair of chords can also be used as a measure of shape roundness. This method is also rotationally invariant.

3.4.4.4 Moments method

Moments are considered at great length in later chapters. The basic method of moments consists of representing the shape by a series of functions based on successive powers of each pixel's x and y axis coordinates [HU 1961]. Koulopoulos rejected the use of moments on the grounds that the moment representation of the shape could not be interpreted. Tout used the second-order moments to obtain a value for the orientation of components and to define a datum for other shape representations. He also included the second-order moments in his shape representation.

3.4.4.5 Slices method

This method calculates the number of pixels of the silhouette that are present in each vertical and horizontal line of the image. The component is then represented by two 'signatures' resembling histograms which can be compared with shapes in the

database. This is rotationally variant, the histograms will change for every different orientation of the shape, thus for recognition to be successful several 'signatures' would have to stored corresponding to differing orientations of the shape. The 'signatures' are not unique, it is quite likely that several shapes will produce the same or at least very similar histograms.

3.4.4.6 Circle measures

Circles of various radii are imposed onto the centroid of the shape which must first be located. The number of times each circle crosses the shape edge is noted and used as a feature to describe that shape. This method is very similar to that of Brezina's radius density method except that the lengths of the radii to be considered are fixed as opposed to the angles at which they occur.

3.4.5 Position of shape centroid

Some of the recognition methods described previously require the shape centroid to be located as it is needed as a reference for radii and circles. The position of the shape is also an important factor if any operation is to be performed upon it. Koulopoulos combined these two requirements as one and used the shape centroid to define its position. The position of the centroid of a shape (provided it is the only shape in the image) is defined in equations 3.1 and 3.2.

$$\text{X Coord} = \frac{\sum_{\text{all}y}\sum_{\text{all}x} f(x,y)\,x}{area} \quad \ldots\ldots\ldots\ldots \text{eq } 3.1$$

$$\text{Y Coord} = \frac{\sum_{\text{all}y}\sum_{\text{all}x} f(x,y)\,y}{area} \quad \ldots\ldots\ldots\ldots \text{eq } 3.2$$

f(x,y) = 1 for pixels within the component outline,

f(x,y) = 0 otherwise.

Note that this requires an inverted binary silhouette.

3.4.6 Matching radii

This is yet another variation on the use of radii and was chosen by Koulopoulos as the basis for his shape representation since it provided an element of orientation assessment. As with Brezina's radius method the radii from the centroid to the edge of the silhouette are sampled at fixed intervals. These radii are then stored and can be compared with other such representations in order to achieve identification. If the comparison between shapes is to be effective then the radii samples must start from the same datum. Koulopoulos' choice of the longest radius of the shape as the datum had a significant flaw in that it is not guaranteed to be unique for any one shape. Tout also used the matching radii approach but modified the method to use the axis of major symmetry as the datum. A silhouette may have more than one axis of symmetry but such shapes can be matched provided the axes line up (an equilateral triangle has three axis of symmetry and can be matched in three different positions).

Note that this method allows reconstruction of the original shape to an accuracy dependent on the number of radii used.

The orientation of a shape is determined as the angle between the datum and the x-axis.

Tout did not find the longest radii of the shape and determined the orientation of most shapes using the angle between the x-axis and that of the major axis of the shapes equivalent ellipse (section 3.5.3.1.). Under certain conditions this was found to be an unsatisfactory method in which case the shape and its prototype were represented by several hundred radii (section 3.5.3.2.) using the x-axis as a datum. The orientation of the shape can then be assessed by finding the number of radial samples through which the shape vector must be turned before it matches the prototype vector.

3.5 The practical application of Koulopoulos' and Tout's work

Tout's system [TOUT 1989] based on the work of Koulopoulos [KOULOPOULOS 1982] used a set of polar vectors and area to describe the shape, the average coordinates of the shapes x and y pixels to determine the component centroid and either the angle of the principle axis of the equivalent ellipse or an enlarged set of polar vectors to give a comparative value for shape orientation.

3.5.1 Centroid position

The centroid of the shape (X, Y) is defined as the average position of all the x and y coordinate values respectively of the pixels that are contained within the shapes outline or silhouette. The centroid coordinates are unique and thus fix the position of the component relative to the system origin.

Initially the shape is represented by its edge points which are extracted from the camera image by hardware using the principles explained in section 2.2.5. The edge points are stored in shift registers until scanning is complete.

Once a shape has been completely scanned it is analysed by the software which begins by reading in the co-ordinates of the edge points from the shift registers and calculating the required centroid position. The contributing values of the pixel positions can be calculated as each scan of the camera is read in by the processor.

3.5.2 Polar vector shape descriptor

Once the centre of the shape is known then the radial distance of each edge point from this centre can be calculated. Since the radii cannot be found until the centroid is known this process can only be performed when a shape has been completely scanned.

A typical shoe shape will have 5,000 edge points so in view of database storage and search time requirements the number of radii must be reduced. The value used for normal operations is 18 radii distributed at 20 degree intervals. The datum used is the direction of the major principal axis as defined by the second order moments of inertia:-

$$\mu_{1,1} = \sum_{x=x_a-X}^{x_b-X} \sum_{y=Y-y_a}^{Y-y_b} F(x,y)\,xy \quad \text{.....eq 3.3}$$

$$\mu_{2,0} = \sum_{x=x_a-X}^{x_b-X} \sum_{y=Y-y_a}^{Y-y_b} F(x,y)\,x^2 \quad \text{.....eq 3.4}$$

$$\mu_{0,2} = \sum_{x=x_a-X}^{x_b-X} \sum_{y=Y-y_a}^{Y-y_b} F(x,y)\, y^2 \quad \text{.....eq 3.5}$$

where F(x,y) = 1 for pixels within the shape.

F(x,y) = 0 otherwise.

The direction of the principal axis relative to the x axis of the co-ordinate system is then given by:-

$$\theta = \frac{1}{2}\arctan\left[\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}\right] \quad \text{.....eq 3.6}$$

(for derivation see e.g. CASE *et al.* 1971)

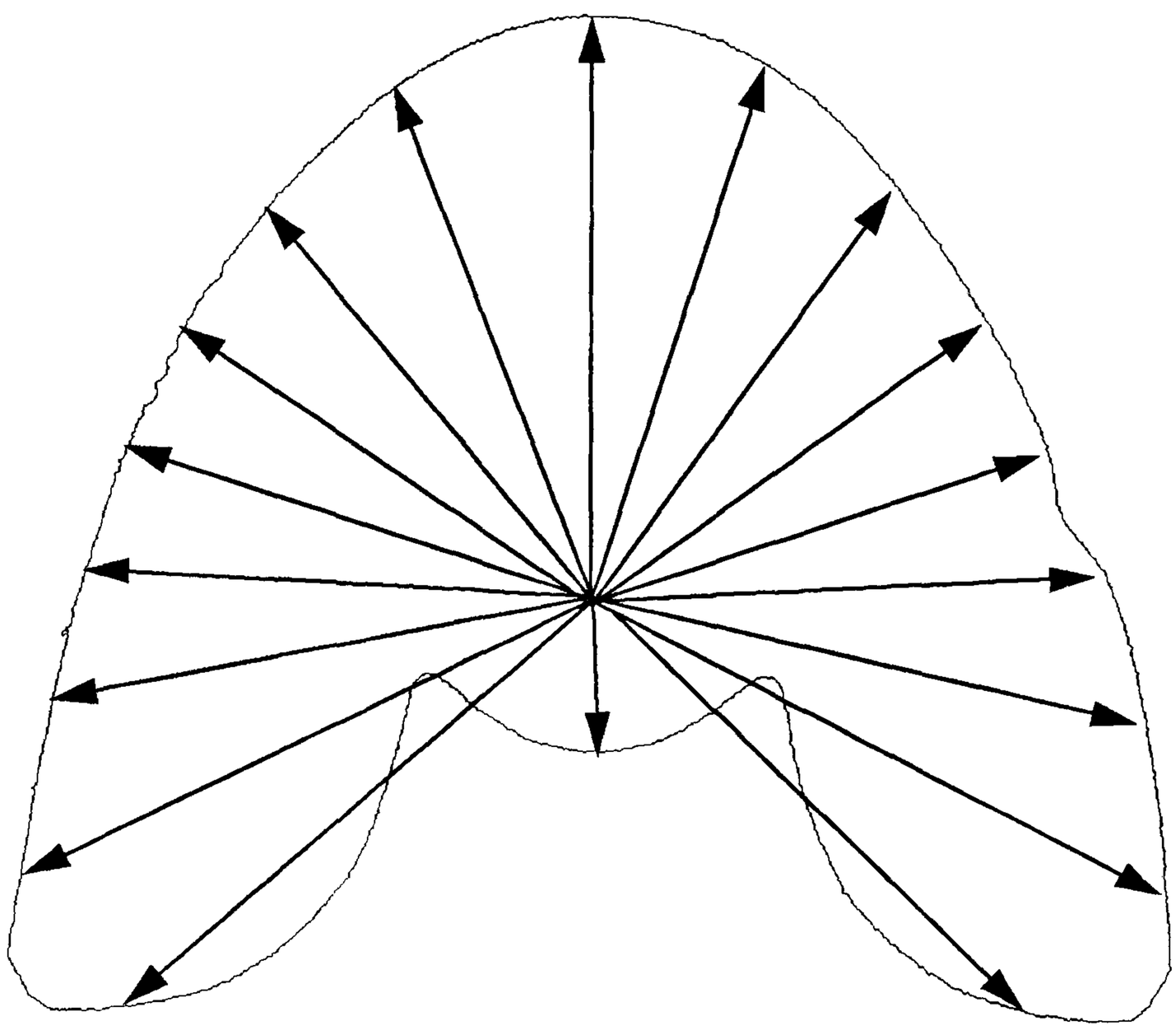


Figure 3.2 A shape described by 18 radii.

Note that if any radii pass through two or more edges it is that which is furthest from the centroid which is used in the representation.

In addition to these 18 radii the area of the shape and the value of its second order moments of inertia are retained as part of the shape representation. Thus the shape is represented by a set of 22 features which are stored in the data base.

3.5.3 Shape orientation

If a pattern is to be stitched onto a component then the orientation of both the prototype and the new component must be known so that any pattern can be adjusted to take account of the difference. The accuracy of the orientation value will determine the standard of the final stitch position.

Tout [TOUT 1989] uses one of two methods depending upon the 'condition' of the shape.

3.5.3.1 Direction of the major principle axis

The direction of the major principal axis as calculated in section 3.5.2 can be used to define the orientation of the shape provided the shape is suitably conditioned. The major principal axis of the shape can be considered as the major axis of the shapes equivalent ellipse.

The equivalent ellipse of a shape is that which has the same area, the same first order and the same second order moments. Equation 3.6 can be used to evaluate the angle of the principal axis of this ellipse with that of the x axis. This angle is directly proportional to the rotation of the component and can normally be used to evaluate its orientation. However as the equivalent ellipse approaches circularity the accuracy of the equation falls away such that patterns are stitched in the wrong place leading to a wasted component.

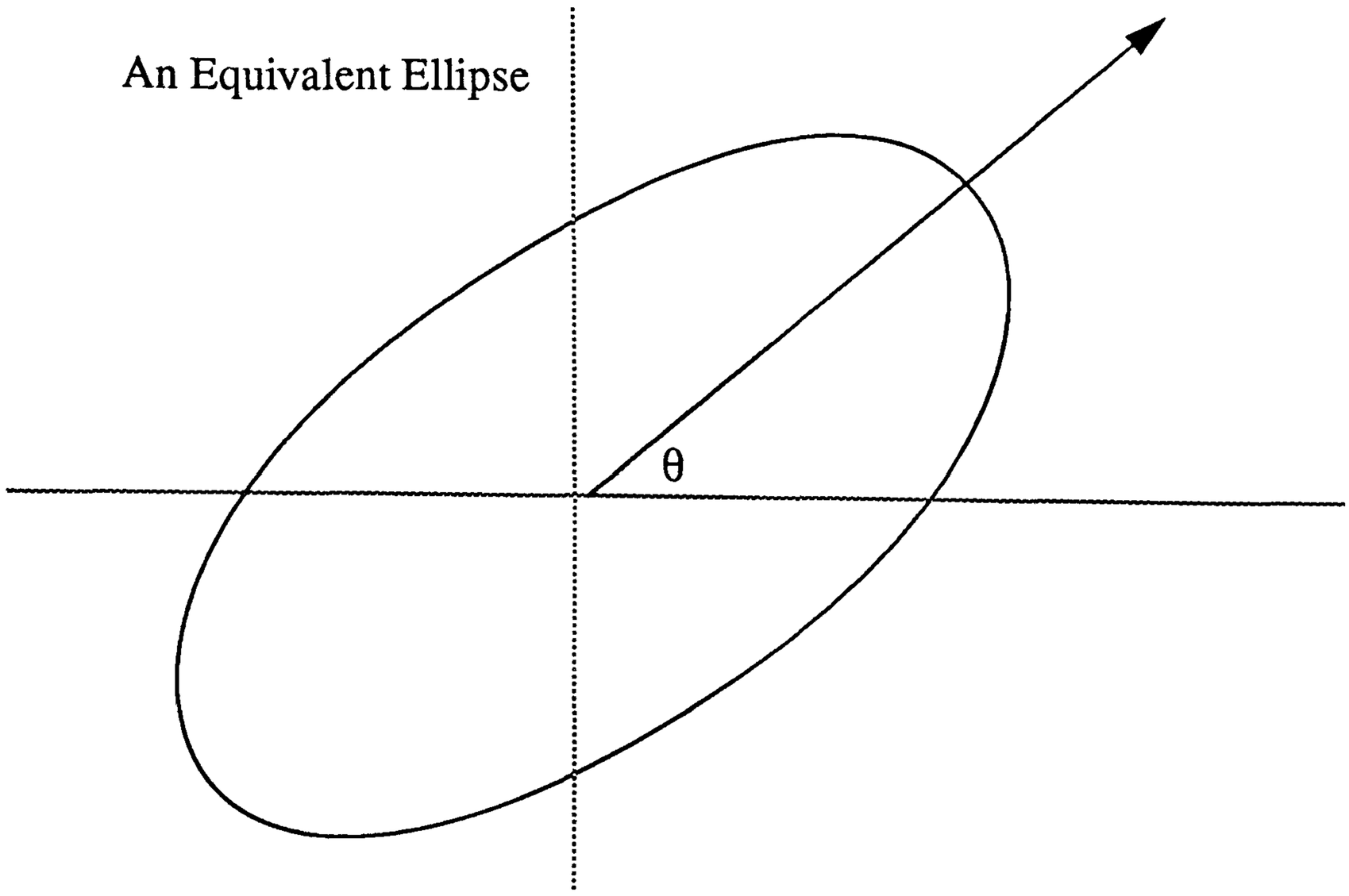


Figure 3.3 An equivalent ellipse.

The use of this equation is fast and easy, thus it is the preferred option.

3.5.3.2 Additional polar vectors

For those components where the equivalent ellipse method is inappropriate the alternative used is to describe the shape using 360 radii from the centroid to the outside edge spaced at 1 degree intervals. To determine the orientation of a shape it must first be identified so that the correlation between the 360 radii of the prototype and the 360 radii of the new component can be investigated. The point of greatest correlation indicates the difference in orientation between the two shapes to ± 0.5 degrees. Interpolation is then used to get the accuracy of ± 0.1degrees required for stitching.

3.5.3.3 Shape 'condition'

The choice between the two methods is determined by the 'goodness coefficient' (GCOEFF) of the shape which has been defined by BUSM [BLOWER 1982] and Tout as;

$$GCOEFF = \frac{\hat{\mu}_{0,2} - \hat{\mu}_{2,0}}{\hat{\mu}_{0,2} + \hat{\mu}_{2,0}} \ldots\ldots\ldots\ldots \text{eq } 3.7$$

The critical value for GCOEFF has been determined experimentally as 0.2. If GCOEFF is less than 0.2 then the shape is said to be ill-conditioned; the result of using equation 3.6 will be unsatisfactory and the additional polar vectors method must be used [BLOWER 1982] [TOUT 1989]. GCOEFF will approach zero as $\mu_{0,2}$ and $\mu_{2,0}$ tend towards the same value, this is reflected by the equivalent ellipse approaching circularity.

3.5.4 Autostitcher training mode

For the system to be able to identify a shape it must first be taught the features of an example of that class. The procedures in 3.5.1, 3.5.2 and 3.5.3.1 for obtaining the polar vectors, area, second order moments, centroid position and orientation are used to produce a typical feature set for that class of shape along with the position and orientation of the prototype. These values are stored in the system database. A joystick is then used to manually control the sewing machine and rollers in order to stitch the required pattern on to the prototype. The co-ordinate points of the pattern with reference to the shape centroid are also stored in the system database.

This procedure is repeated for any additional prototypes.

3.5.4.1 Training for ill-conditioned shapes

As each prototype is presented the value of GCOEFF is calculated using equation 3.7. The required minimum and maximum values of the moments are found as part of the routine to determine the major and minor axes. If a prototype with a GCOEFF of 0.2 or less is presented then the system will further describe that component by a series of 360 polar vectors at 1 degree spacing using the same datum as the standard 18 radii description. The additional radii are also stored in the database.

3.5.5 Autostitcher recognition mode

An un-sewn component is fed onto the machine and taken up by the rollers. The system proceeds to extract a set of 22 features as described in 3.5.1, 3.5.2 and 3.5.3.1. The value of GCOEFF for this unknown shape is calculated and if it is less than 0.2 a further set of 360 radii is obtained.

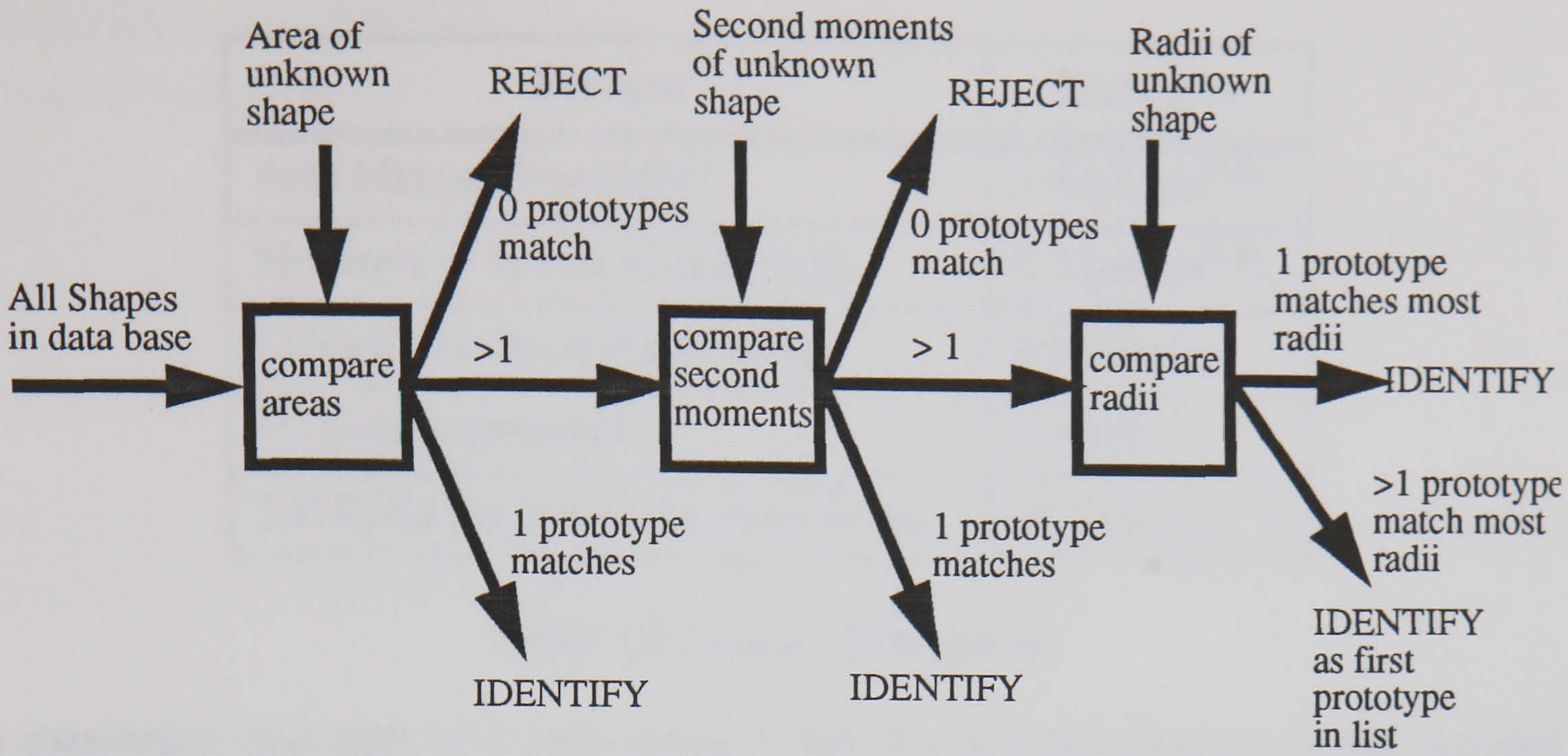


Figure 3.4 Tout's identification process for the Autostitcher.

Identification of the shape is a multi-stage process to minimise database search time and begins with a primary filter based on comparison of the area of the unknown shape with each prototype stored in the database. Prototypes that have an area within a given tolerance (table 3.1) of the unknown shape proceed to phase 2.

The second phase compares the moments of inertia and passes only those prototypes that match within a given tolerance. For most shapes only one prototype will match after these two stages and can be used as the identification of the unknown component.

If more than one prototype passes phase 1 and phase 2 then identification is concluded by matching within a given tolerance the 18 radii of the unknown shape and the remaining prototypes. The prototype with the best number of matching radii is taken as the identity of the component subject to the maximum mismatch specified below. If the value of GCOEFF is greater than 0.2 then the orientation of the component can be determined using equation 3.6 and each set of 18 radii need be compared only once. If the orientation cannot be found in this way then the radii sets must be compared in each of 18 possible combinations.

Feature	Tolerance
Area Measurement (root)	± 4 (pixels$^{0.5}$)
Moments of Inertia (fourth root)	± 4 (pixels$^{0.5}$)
18 Radii for shape identification	± 5 (pixels)
Maximum mismatch	5 (radii)
360 Radii for orientation assessment	± (5 pixels)

Table 3.1 Feature Tolerances.

A maximum mismatch of 5 radii requires that the feature-radius set of the scanned shape and the best match prototype must have no more than 5 radii which do not match within tolerance, otherwise the scanned component will be rejected.

Note that if two or more components have the same radii then the first shape encountered in the database is used for identity purposes.

If the system fails to come up with an identity for a shape the operator will either reject the component or re-present it in a different position.

For those components with a GCOEFF greater than 0.2 the orientation will already have been found. For the remaining shapes the orientation is found by correlating the 360 radii of the component with those of the prototype. The prototype radii are then shifted through 1 degree and the correlation repeated. After 360 such comparisons have been performed the orientation of the component compared with that of the prototype can be determined by the position showing greatest correlation. Interpolation is then used to increase the accuracy of the measure.

The corresponding sewing pattern can then be shifted from the prototype centroid to the new component centroid and rotated by the difference in angle prior to stitching. In order to rotate the sewing pattern around the shape centre the system uses the following relationships;

$$x' = x\cos(\theta) + y\sin\theta \quad \text{.....eq 3.8}$$

$$y' = -x\sin\theta + y\cos\theta \quad \text{.....eq 3.9}$$

Where (x, y) are the original coordinates of the pattern transformed to the new centroid, θ is the orientation difference between the prototype and identified components and (x', y') are the new coordinates of the pattern.

The rollers can now be controlled in order to stitch the correct pattern in the correct place at the correct orientation.

When stitching is complete the component is ejected from the system.

The system now awaits the next component which may be a new component requiring stitching or an additional prototype to add to the database.

3.5.6 Performance

Several parameters are used to measure performance. These are speed of operation, accuracy of stitched pattern, recognition success rate and database storage size.

The time to process a component will depend on the component size, the database size and the speed of the processor. The best available estimate of the polar system used by Tout is 3.3 seconds for the scanning and recognition of an average component. For the purposes of algorithm comparison this would have been better expressed in terms of 'processor clock cycles', but such information is unavailable.

The accuracy of the system is all important if quality is to be maintained. The original specification as set out in 3.1 calls for an accuracy such that no stitch is more than 0.25mm away from its correct position.

Tests on Autostitcher systems have demonstrated good repeatability. Providing a shape is correctly recognised then the required pattern is stitched onto the component such that each stitch is within 0.2 mm of the taught position. This is a result of the accuracy of the positional evaluation and the two methods used for orientation assessment. Note that if the value of GCOEFF is less than 0.2 and the second moments method is used then the position of stitches does not give the required quality.

Recognition performance is the single most important parameter, if the system can not recognise shapes then it can not perform its designated task. There are three possible outcomes from the recognition stage of the process:-

(a) Component correctly identified.

(b) Component unidentified and rejected by system.

(c) Component recognised incorrectly as being a member of a different class.

The third possibility is a substitution error and is the most damaging since the wrong pattern will be stitched to the wrong shape leading to scrapped materials. Thus substitution must be avoided. Errors of type (b) are rejection errors and normal lead to a component being re-presented for a second attempt. Rejection errors are costly in terms of time but cause no wastage of material. If a component is rejected then it may be faulty, thus the system incorporates some quality checking.

The recognition performance of the system can be evaluated in terms of the relative proportions of substitution, rejection and correct recognition. The relative weightings of these measures can be altered by adjusting the tolerance values of the various comparison stages. For the values of tolerance as given in table 3.1 Tout's evaluation for the performance of the polar system was:-

Rejection rate of 0%.

Substitution rate of 2.3%.

Correct recognition rate of 97.7%.

These figures are from an experiment by Tout [TOUT 1989] involving 1,249 components based on the full range of component type likely to be found in a standard shoe factory.

It is important to note that the evaluation of recognition will depend on the sample universe used to test the machine. Obviously a set of shapes which are easily distinguishable will give an optimistic result, while a set of difficult shapes will give a pessimistic evaluation. Ideally the results must be gathered over time as the universe varies with changing styles and designs.

The database size will vary with the number of components stored and also the proportion requiring the additional 360 radii description. From Tout's [TOUT 1989] discussions on the structure of the expected universe a typical 20,000 component system would require, on average, 7 Mega Bytes of storage space.

3.5.7 Effects of changing the resolution

Jinabhai [JINABHAI 1992] evaluated the effects of changing the resolution on Touts' system and concluded that the rejection and substitution rates could be improved by increasing the resolution of the image. The evaluation was performed using scaled

components rather than different camera set ups. Although resolution can be increased by lowering the camera height this has the effect of reducing the scan width and the size of components that can be scanned. Lowering the camera will also worsen the apparent width error due to material thickness as discussed in section 2.2.3.6. The ideal solution of a larger array camera was too expensive and is yet to be implemented.

3.6 Summary

The present system based on the work of Koulopoulos and Tout does not achieve the performance as set out in the original specification (section 3.1).

The edge of a silhouette shape contains the greatest proportion of information about its identity. The use of a limited number of radii to sample the shape boundary means that significant information is lost and as the size of the shapes increases then so this problem is compounded due to the edge samples spreading out. In order to overcome this problem methods must be used which respond to a greater proportion of the pixels contained within the silhouette, either on the boundary or globally over the whole shape. It is, however, important to avoid the problems of noise associated with boundary pixels.

On the question of component orientation it is apparent that Tout's methods have achieved the required standard, but the processing of components with a low GCOEFF value is intensive. The power of modern day processors means that this is not a significant problem but one which never the less might be addressed while investigating improved recognition performance.

CHAPTER 4

SHAPE REPRESENTATION

4.1 Introduction

A thresholded, binary image of 2048 pixels by 2048 pixels contains nearly 4.2 Mega bytes of information. If such images were to be stored in this raw state then a 20,000 component data base would require 840 Giga bytes of memory or disk space.

The identification of a component would involve the comparison of each image on a pixel by pixel basis. Such image comparison would fail for components in different positions and in different orientations. The following sections describe the principles of feature extraction whereby the image is described only by some characteristics which are invariant to the actual position and orientation of the component. This set of characteristics or features requires much less storage space and allows comparisons between components in different positions and orientations. In the worst case when the features can take on only a binary value a data base of N components will require Log_2N bits for each component. A 20,000 component data base would require a total of 300 Kilo bytes which is a very small fraction of the 840 Giga bytes needed to store every image.

4.2 Multi dimensional feature space

Entities described by a set of values or features can be represented by a point within a multi dimensional hyper-space known as the feature space. Each feature corresponds to an orthogonal axis. The feature values for each shape lie on the corresponding axis thus the intersection of these values represents the shape. The feature values can be considered as a set of coordinate points describing the shape, object or entity.

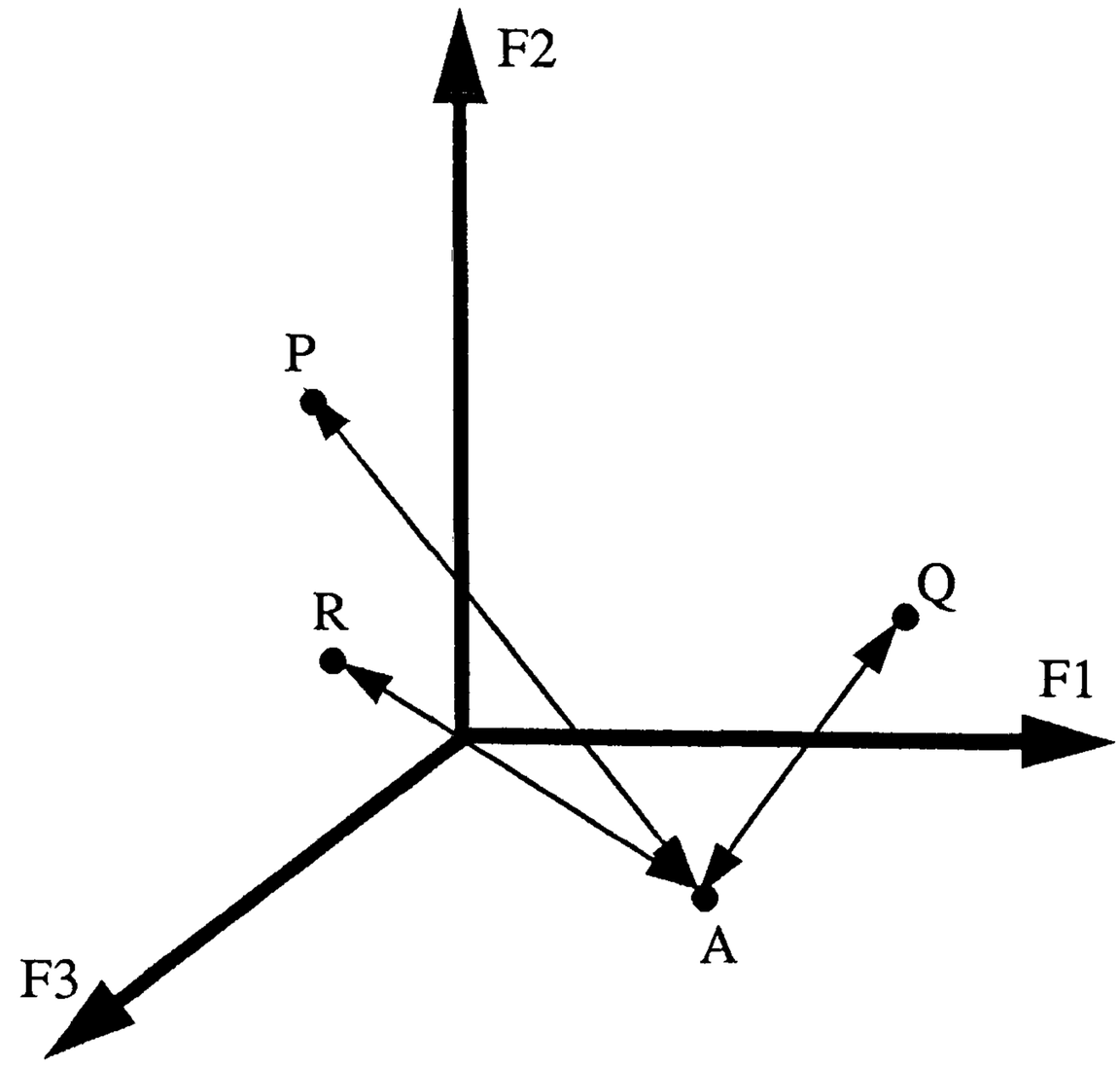


Figure 4.1 Points in a 3 dimensional feature space.

The distance between two shapes as represented by the separation of their two hyper-space points (the Euclidean distance [e.g. BROWNE 1986]) is a measure of the similarity of the shapes. The smaller the distance, the more similar the shapes.

A recognition system based on feature representation operates by assessing the difference in position between a point representing the unknown shape and the points representing each of the prototype shapes in the data base. The unknown shape will normally be classified as belonging to its 'nearest neighbour'. A necessary process in this method is the establishment of the prototype points during the teaching phase.

4.2.1 Feature vector

The set of feature values describing a shape are acting as a set of coordinates, pointing to one location in the hyper-space. For this reason they can also be considered as a vector representation and referred to as a feature vector.

4.2.2 Deterministic and statistical properties

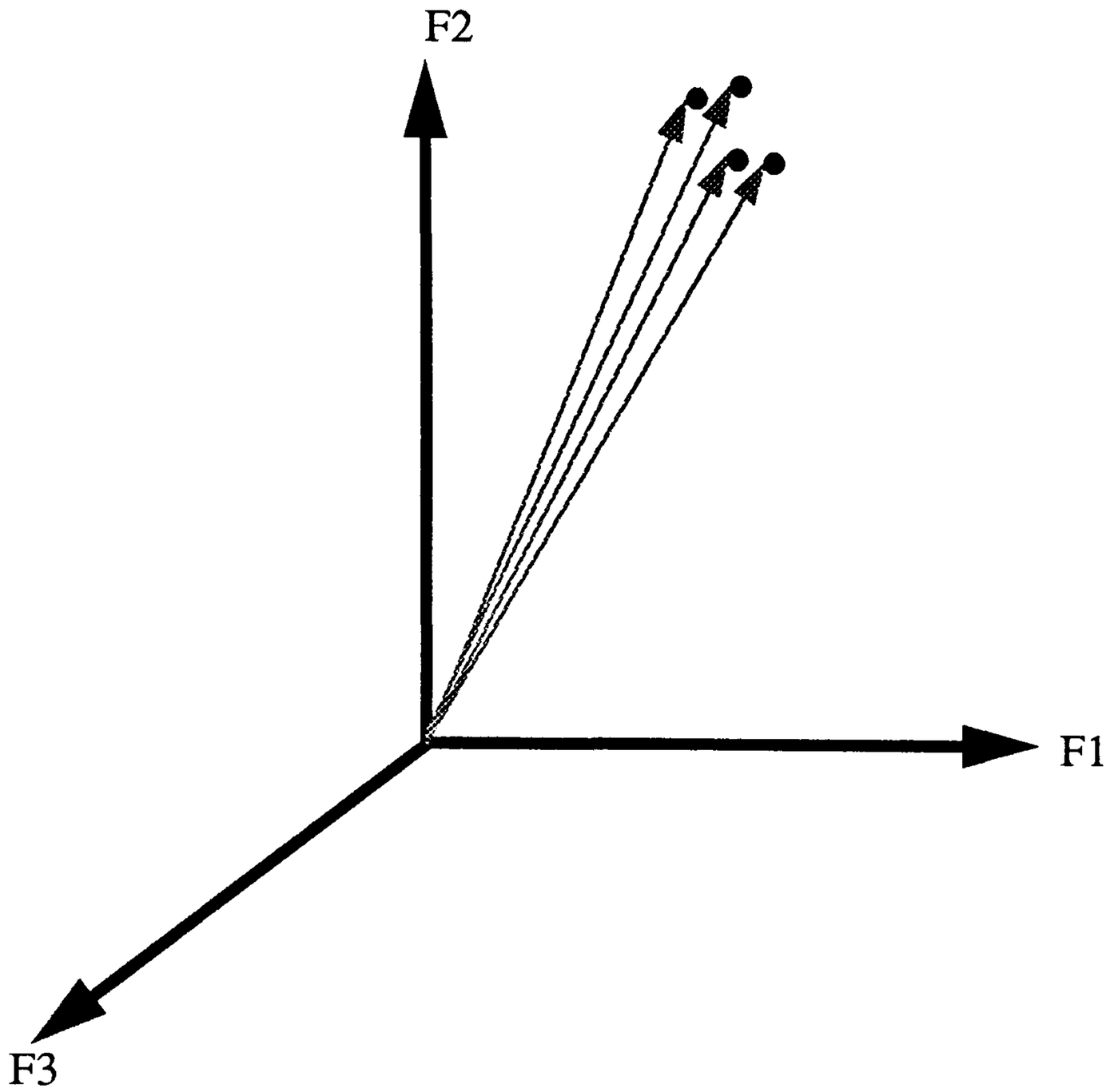


Figure 4.2 Several points in 3 dimensions from images of 'identical shapes'.

As an object is moved, rotated or viewed from a different position the image will change slightly due to noise caused by quantisation, light variation and distortions in the object. On extraction of feature vectors from these different images of the same shape the point in the feature space may or may not vary depending on the features used. Similarly if different examples of the same shape are used the corresponding points in the feature space may or may not vary.

If all examples of a class of shape produce a feature vector describing exactly the same point in the feature space then that class is deterministic. If the feature vectors vary such that a region of the feature space is covered then the class is statistical.

Very few problems will be strictly deterministic since this would normally require identical images of identical components. Most vision problems contain some element of statistical behaviour. The identification of shoe components is no exception. At first glance the problem appears to be deterministic. The components are cut from the raw material with knives bent to the required shape as required by the shoe design. The identification of a shape is theoretically simply a case of matching shapes and knives, with any shape having no match being rejected. In practice the cut shapes and knives vary. The process of identification now becomes one of matching an unknown shape to

a region within the feature space. If two shapes are very similar then these regions in the feature space could overlap leading to recognition difficulties.

4.2.3 Distribution

The dimensions of the feature space will depend on the features used, with the feature vector for a shape lying anywhere within that space. If a single training example algorithm is used then only one vector can be used to determine a prototype region within the feature space.

4.2.4 Clustering

A typical shoe factory produces 20 different designs of shoe with 20 different components each with left/right variations. In addition each shoe is produced in 20 half size variations. This leads to generically similar components in terms of shape and area. The resulting effect on the feature space can be seen as groups of components which exhibit similar features clustered together in groups or strings.

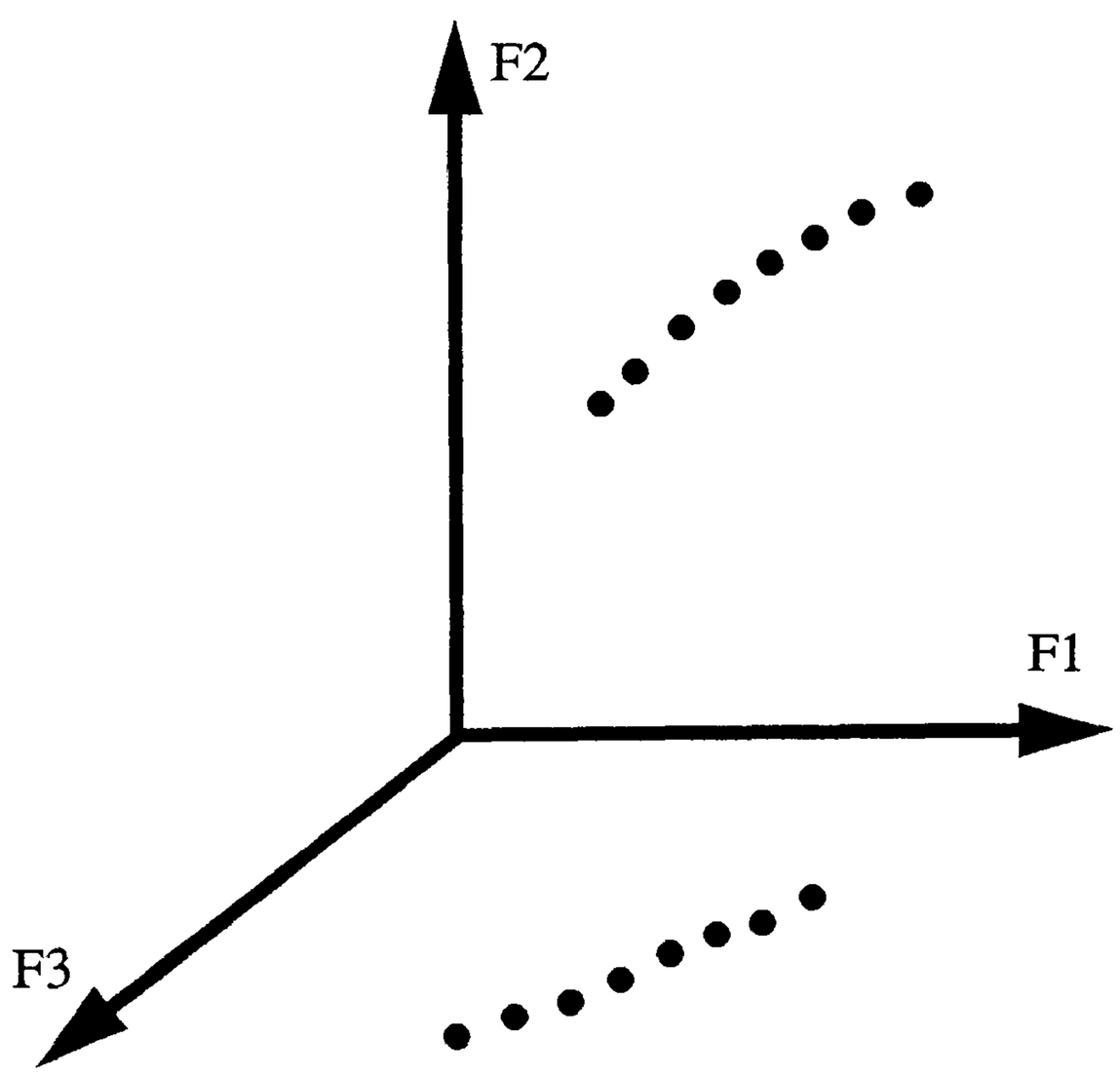


Figure 4.3 Clusters in the feature space.

The clusters are normally sparsely distributed so that the task of identification becomes a two stage process of locating the correct cluster followed by the correct prototype within the cluster. This process can be seen in Tout's identification algorithm (see Figure 3.4) which uses area as a primary filter to locate the cluster of prototypes with similar area followed by moments and radii to identify the actual prototype.

4.3 Identification by feature vector

Norton-Wayne proposed an identification algorithm [NORTON-WAYNE 1982] based on the Hamming code principle [e.g. GALLAGER 1968] used in digital signal processing techniques for correction of errors. The method involves dividing the feature space into a discrete set of regions by allocating each prototype feature a defined tolerance. The greater the tolerance the bigger the prototype region and the lower the resolution of the feature space.

The identification of a shape requires the calculated feature vector to be compared with each prototype vector in the database on a feature by feature basis. For every feature of the unknown shape falling within the tolerance of the prototype in question the number of matching features is increased by one. Thus for each prototype the total number of features falling within tolerance is calculated and can be compared with the values for the other prototypes. The prototype with the highest score is the best match.

It can be visualised that for greater tolerance (low feature space resolution) component features are more likely to match prototype values, leading to lower numbers of rejections and higher numbers of substitutions. Lower tolerances (high feature space resolution) will have the opposite effect as illustrated in figure 4.4.

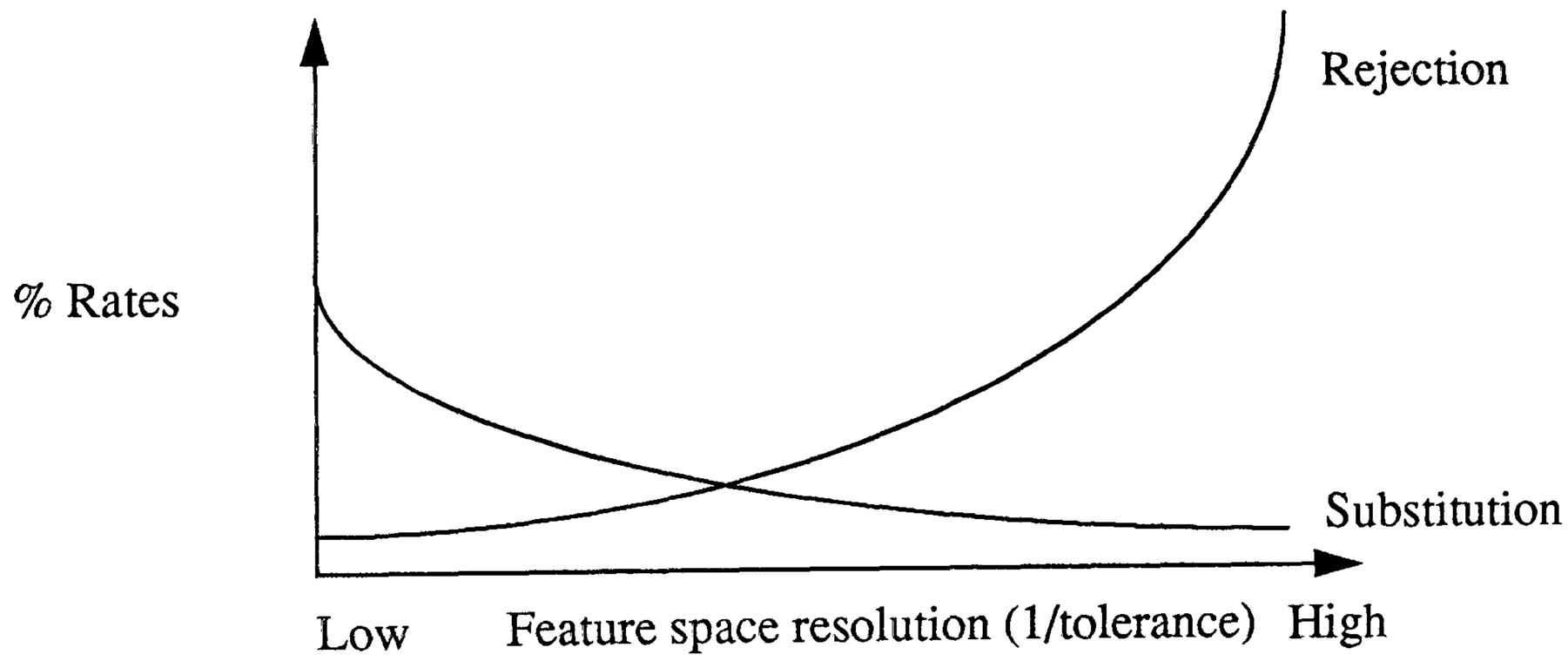


Figure 4.4 Variation of rejection and substitution rates with tolerance.

Although the prototype with the highest number of matching features is the best match, it is not necessarily the correct one. For a system with say 18 features it would be unlikely that a best match of 5 features represents a good chance of correct identification. To protect against substitution errors a minimum number of matched features must be obtained to confirm component identity.

4.3.1 Minimum distance

To reduce confusion between classes the features should be chosen so that more than one feature differs between each class. This separation is known as the minimum distance between classes. The number of features used in a system will depend on the minimum distance required which will in turn depend on the accuracy with which the features can be determined.

4.3.2 Number of features required

If a single feature F_1 can be split into N_1discrete levels;

1, 2.....N_1.

Then N_1 classes can be distinguished with a minimum (and maximum) separation of one feature. For two features having N_1 and N_2 discrete levels the number of classes that can be separated is N_1N_2 since the feature space can be resolved into N_1N_2 regions. If the feature space is multidimensional, having P features with N_1, N_2,.....N_p discrete levels corresponding to P axis then the total number of regions that can be resolved is given by α;

$$\alpha = \prod_{i=1}^{P} N_i \text{.....eq 4.1}$$

If $N_1=N_2=N_3=$.....Np=N then

$$\alpha = (N)^p \text{.....eq 4.2}$$

thus

$$P = \log_N \alpha \text{.....eq 4.3}$$

The number of features of N discrete levels required to partition a feature space into α regions rises logarithmically with α.

The choice of feature will obviously effect the value of N_p. A feature which is constant throughout the database will not aid the decision process. A feature must have at least two different values somewhere in the database to be of any use. Thus the worst case in

terms of minimum number of features required is given by N_p=N=2 (binary) for all features. The number of features needed is

$$P = \log_2\alpha \ldots\ldots\ldots\ldots \text{eq } 4.4$$

If N is greater than 2 then the required number of features will be less;

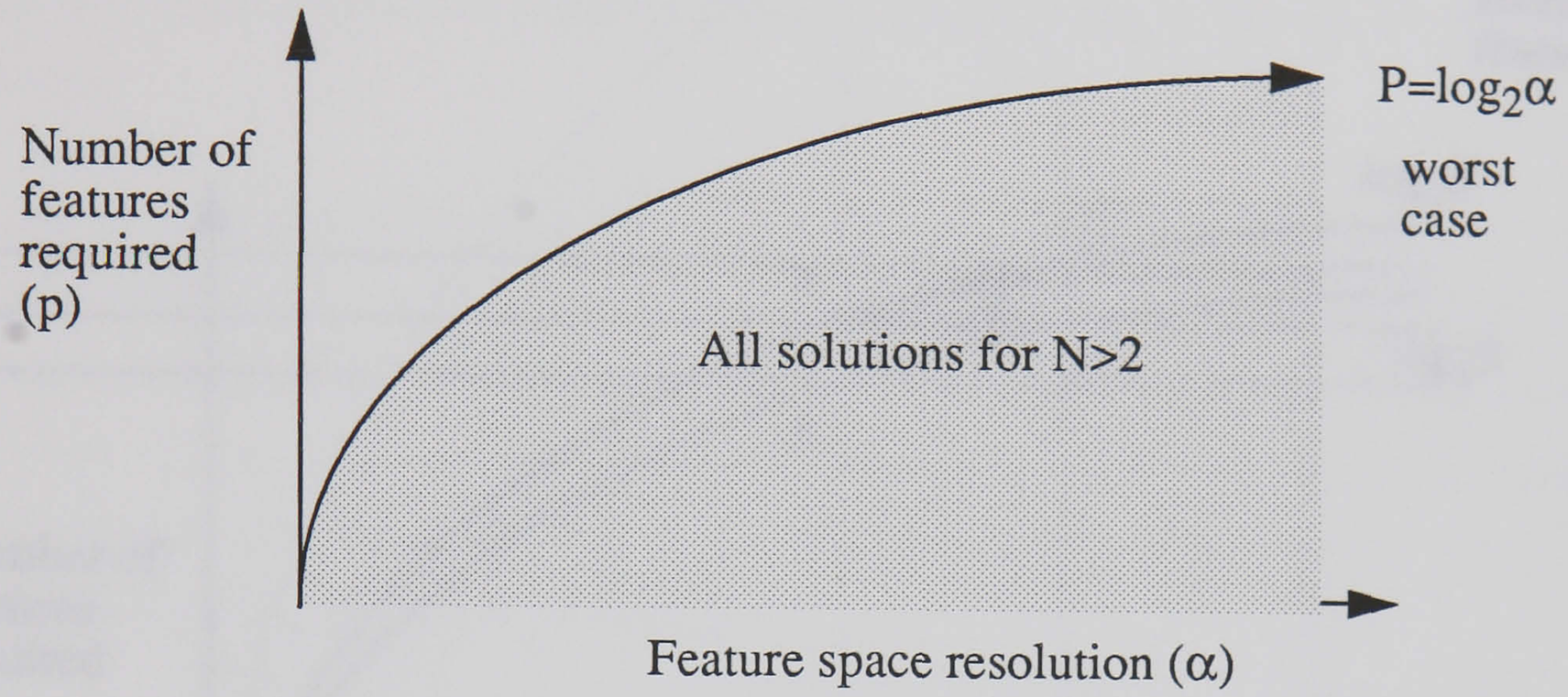


Figure 4.5 Variation of required features with feature resolution.

4.3.2.1 Feature limit

If the minimum number of features is exceeded then the feature vector will contain redundant values. The redundancy will improve the recognition performance since it allows for the fact that some feature values will be corrupted with noise and yet the feature vector will still contain the minimum number of matching features required for identification. It is not, however, simply a case of applying as many features as possible. Hughes [HUGHES 1968] has shown that too many features lead to a fall off in performance.

4.3.3 Manipulating the feature vectors

The choice, number and resolution of the features chosen will affect the system's recognition performance, speed and database storage requirements. The features chosen will depend on the accuracy of the available equipment, the speed required, the storage available and the preference for substitution or rejection errors.

If a system requires β classes to be identified and has α regions then $\alpha >= \beta$. If $\alpha = \beta$ then every region of the feature space is assigned to one class of shape with the result that no shape can be rejected. This is unlikely for a practical system such as shoe upper stitching. A more likely occurrence is $\alpha >> \beta$. This gives rise to a system with many redundant areas of the feature space and thus regions available for rejection. The selection of features can be optimised to suit the preferred/available operating conditions.

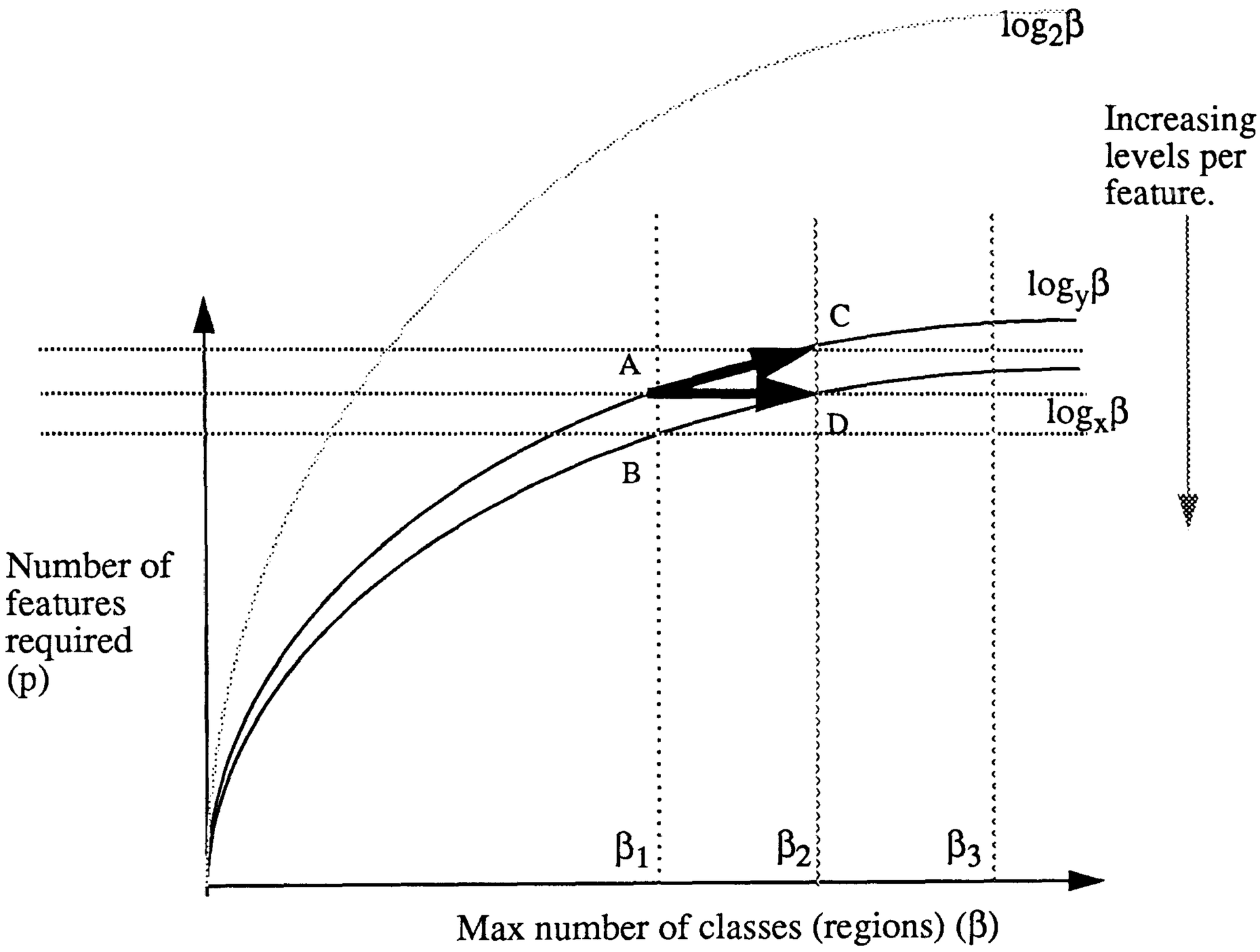


Figure 4.6 Operating regions.

In figure 4.6 the two lower curves illustrate the number of features required to resolve the feature space into β regions when the features can be divided into x and y distinguishable levels. If the problem involves β_1 classes and a solution at point A is used the number of regions within the feature space will be exactly β_1 and the rejection rate can only be zero. To obtain redundancy the resolved zones in the feature space must be increased either by increasing the number of features (moving from A to C) or increasing the resolution of each feature (moving from A to D). In practice a hybrid of the two is likely to be used giving an operating point in the region ACD.

If the new number of zones in the feature space is β_2 then β_2-β_1 is the number of unassigned zones in the feature space and the redundancy can be defined as;

$$Redundancy = \frac{\beta_2 - \beta_1}{\beta_2} \times 100\% \ldots\ldots\ldots\ldots \text{eq } 4.5$$

4.3.4 Optimal feature selection

The optimal feature set will be that which achieves the required minimum distance with the minimum number of features thus reducing storage requirements and database search time.

Given all the features which it is reasonable to extract from an image it is possible to estimate the optimal set by exhaustive experiment on an example set of data. This experiment would require each possible pair of example shapes to be compared for each subset of the total feature set and the number of differing features noted. The optimal set can then be estimated as the one giving the greatest difference between shapes with the least number of features.

Browne and Norton Wayne [BROWNE 1986] reject such exhaustive searches on the grounds that if there are N possible features then the number of subsets to be investigated will approach N!, requiring an overwhelming number of tests. This considerably overestimates the possible subsets which can in fact be shown to number only 2^N by considering the combinations and not the permutations of features (If N = 20 then N! is around $2.4e^{18}$ and 2^N is around $1e^6$). While this is still a large value even for small numbers of features it is possible to evaluate the subsets in search of the optimal feature vector. If there are p shapes in the test set then the total number of pairs that need to be compared is p(p-1)/2 giving a total number of tests of $2^N p(p-1)/2$.

4.4 Criteria of feature selection

There are many features which can be used to represent shape. Those that where considered initially by Tout and Koulopoulos are discussed in chapter 3 while those considered as part of this research are presented in chapter 5. The choice for selection of features depends on several criteria;

- Maximise Class Separation
- Minimise Data Storage Overheads

- High Speed
- Noise Immunity
- Invariance to Rotation
- Invariance to Translation
- Minimise Substitution and Rejection Errors.

4.5 Rejection, substitution and failure rates

There are three possible outcomes from the process of identifying a shape; Reject, Substitute and Correct identification.

For correct identification the features of the unknown component must lie within the boundaries allocated to the true prototype. The shape will then be identified correctly, that is be matched with the required prototype.

When a shape is substituted it is allocated the wrong class description. This is due to the unknown shape's features falling outside of the true class boundary but within the boundary of another, usually adjacent, class.

If the features of a shape do not fall within any prototype region then the shape can not be identified and is rejected.

Substitution and rejection may be the result of a poor feature representation for either the unknown shape or the prototype caused by poor imaging and noise. Alternatively the shape may have been distorted or manufactured wrongly. An on-line operational machine can not distinguish a correct result from a substitution but it can obviously detect a rejected shape. The consequences of a failure to identify correctly will vary according to the subsequent processes. Substitution will generally lead to damage to the original component, a poor quality finished product and possible machinery damage. Rejection usually costs only the time to re-present a component in order to obtain a revised set of features.

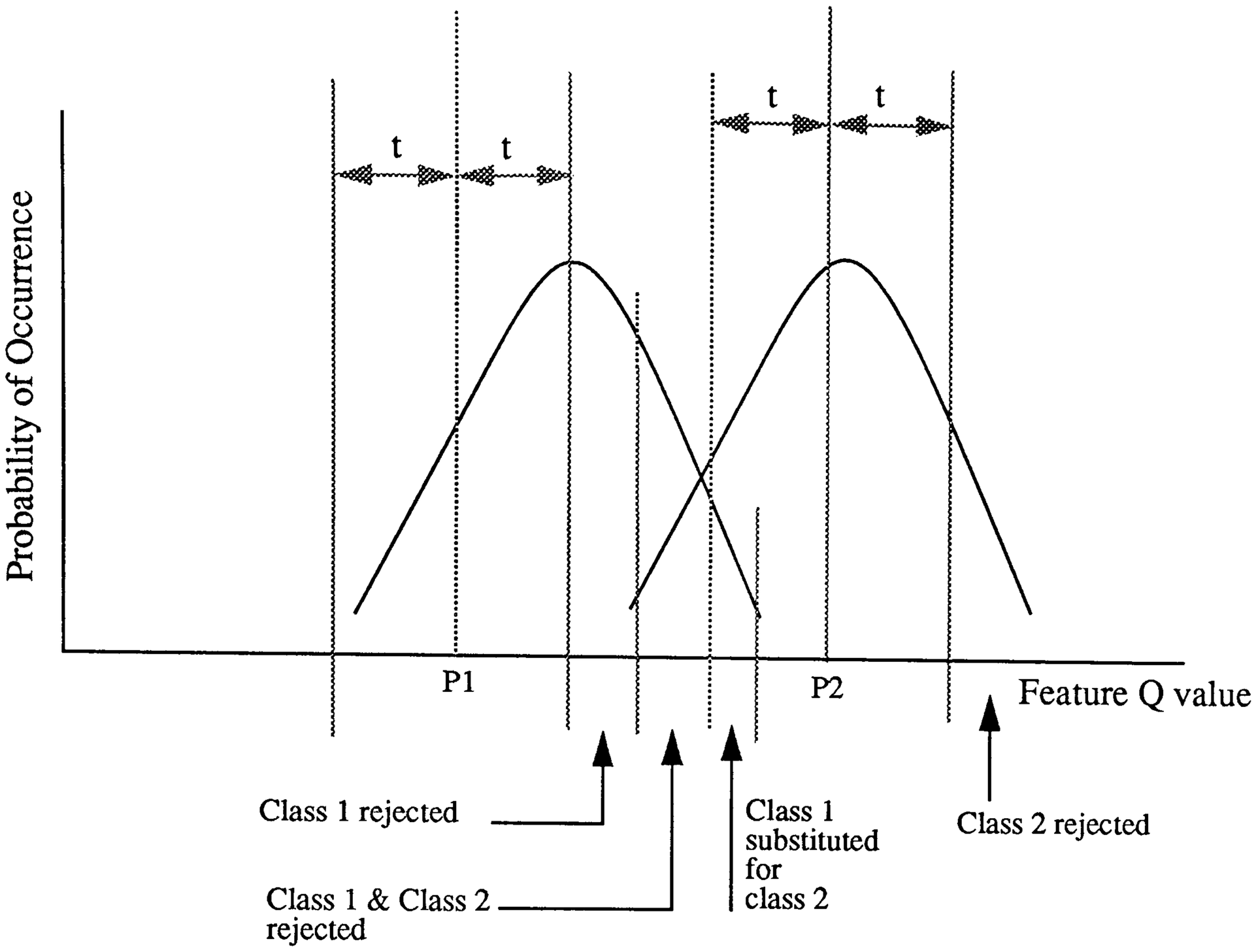


Figure 4.7 Feature variation between two classes.

Figure 4.7 illustrates how feature Q might vary for two classes of component. The system is trained on prototype P1 for class 1 and prototype P2 for class 2. In the case of class 2 the prototype has a feature Q value near to the mean of the population. This is not the case for P1 which has a feature Q value on the lower side of the distribution. In each case any new shape will match the respective prototypes if its feature Q falls within the region $P_{1or2} \pm t$. When examples of class 1 are compared with prototype 1 half of them will fail to match in terms of the Q^{th} feature. There are three ways of reducing the instances of such rejection:

- Increase the tolerance value, t.

- Reduce the spread of the feature values (e.g. filtering or improved lighting).

- Obtain a better prototype value by taking more samples.

The operating point of the system i.e. the relative proportions of rejections and substitution can be set by the selection of suitable feature descriptors and tolerances within the database searches.

4.5.1 Binomial description of failure rates

A component is either identified correctly or incorrectly and as such the process can be considered as a binomial probability.

Consider a system which represents components by N features and will reject any component match if L features or more are mismatched. If the average probability (see appendix VI) of a feature being mismatched is p(0) then the instances of failure could be estimated thus:

The number of combinations of L features in N is given by;

$$\frac{N!}{L!\,(N-L)!} \ldots\ldots\ldots\ldots \text{eq } 4.6$$

and the probability of any L features being in error is;

$$p(0)^{L}(1-p(0))^{N-L} \ldots\ldots\ldots\ldots \text{eq } 4.7$$

thus the probability of a match rejection due to L mismatches is;

$$p(L) = \frac{N!}{L!\,(N-L)!}p(0)^{L}(1-p(0))^{N-L} \ldots\ldots\ldots\ldots \text{eq } 4.8$$

A match will be rejected if L, L+1, L+2,.....,N. features are mismatched. The overall probability of a mismatch will be;

$$\text{Failure}\,Rate = \sum_{l=L}^{N} \frac{N!}{l!\,(N-l)!}p(0)^{l}(1-p(0))^{N-l} \ldots\ldots\ldots\ldots \text{eq } 4.9$$

If a component is not correctly matched with its prototype then it will either be rejected completely or substituted for another class. If substitution occurs it is normally with an adjacent, similar class. The method of substitution can be considered as not being rejected by this false class. For this to happen N-L+1 or more features must match with

the false class. If q(0) is the probability of a feature in class x matching a feature in class y then the probability of substitution between any two classes is;

$$p(sub) = \sum_{B=N-L+1}^{N} \frac{N!}{B!(N-B)!} q(0)^{B} (1-q(0))^{N-B} \quad \ldots \text{eq } 4.10$$

The substitution rate will be the average of this value for each pair of classes in the system. However q(0) will vary from class pair to class pair (p(0) is fixed for the system - assuming the feature distribution is the same for all classes) and will approach zero for all class pairs apart from those which are very similar. To calculate the substitution rate for a shape involves the average for equation 4.10 over the immediate neighbours. The substitution rate for the system can be estimated from an average of such rates.

The rejection rate can be estimated using:-

FAILURE RATE = SUBSTITUTION RATE + REJECTION RATE

where;

The failure rate is the proportion of components with L or more features incorrect, the substitution rate is the occurrence of components with L or more features falling inside the tolerance bands of other shapes and the rejection rate is the proportion of components with L or more features falling in no tolerance band.

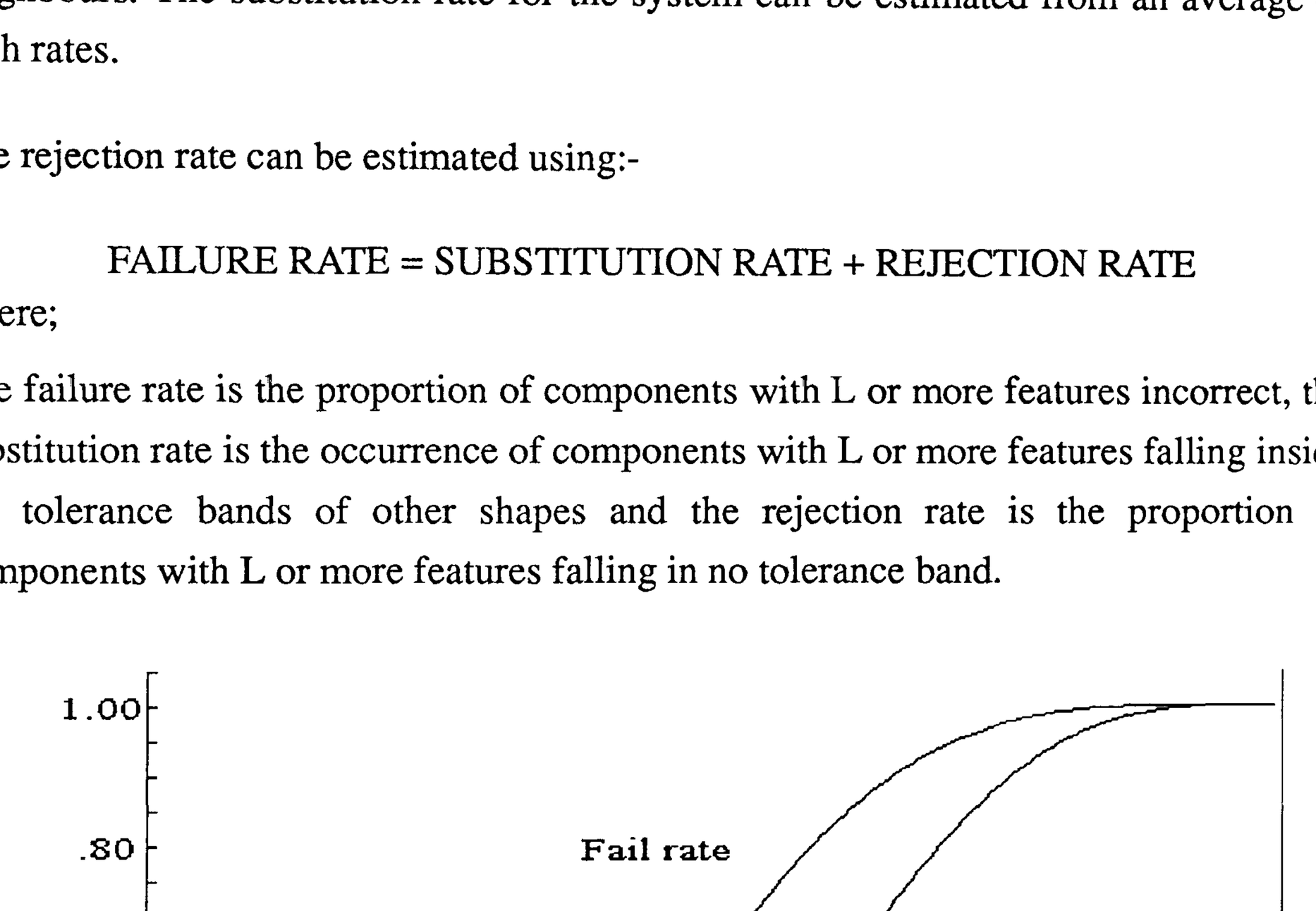


Figure 4.8 Failure rates and substitution rates against values of q(0) and p(0).

4.6 Summary

Describing a component by a feature vector reduces image storage requirements, reduces processing times and can, if suitably selected, remove effects such as orientation, translation and lighting which may hinder the identification process. The feature vector description can also include redundancy, allowing identification even though some features may be corrupted with noise.

This chapter has discussed some theoretical aspects of a system representing shape form using features:- How components are represented, how they are identified and the errors associated with such a system. Chapter 5 will consider some actual feature measures while chapter 6 and chapter 8 will apply a system of feature representation to the problem of shoe component identification.

CHAPTER 5

METHODS OF SHAPE DESCRIPTION AND THEIR USEFULNESS FOR DETERMINING IDENTIFICATION, ORIENTATION AND POSITION

5.1 An evaluation of methods

Many methods have been developed for the automatic interpretation of silhouette images and objects within those images. The hardware, software and algorithms used will depend on the situation and the requirement of each task.

In the case of shoe components they are viewed from above and lit from underneath. A threshold is applied to the image seen by the camera and a binary silhouette is generated, which is then inverted so that the pixels forming the component are represented by binary '1's and those forming the background by binary '0's. The formation of the binary silhouettes considerably simplifies the image which now consists of large groups of '1's and '0's as opposed to a grey scale image which is made up of many small groups of values.

Further to those methods investigated by Koulopoulos and Tout and discussed in chapter 3 the following methods were considered.

5.1.1 Edge representations

Once the silhouette has been obtained then its perimeter can be tracked and recorded using a variety of methods. The method used will depend on the predictability of the outline and the memory available. Edge representations are invariant to translation but not rotation and are prone to the noise problems associated with the perimeter (see

3.4.4.1). In each case identification is achieved by a comparison between sets of numbers that describe the perimeter.

5.1.1.1 Chain code and crack code

A silhouette shape is described completely by its boundary and each pixel on the boundary has exactly two neighbouring pixels which are also on the boundary. Only eight relationships are possible between pairs of neighbouring cells, if both direct and diagonal transitions are acceptable. This leads to the description of a silhouette's boundary by the chain code [e.g. DAVIES 1990]. It is a sequence of numbers in the range 0 to 7, which describes the shape by the position of the edge pixels relative to their closest neighbour (figure 5.1). The resulting descriptor is independent of position but still contains orientation information. In order to compare two shapes their chain codes must be compared, with each edge pixel in turn used as a starting point. The point of greatest correlation can be used to indicate relative orientation. If the coordinates of the first pixel are known then the silhouette image can be reconstructed.

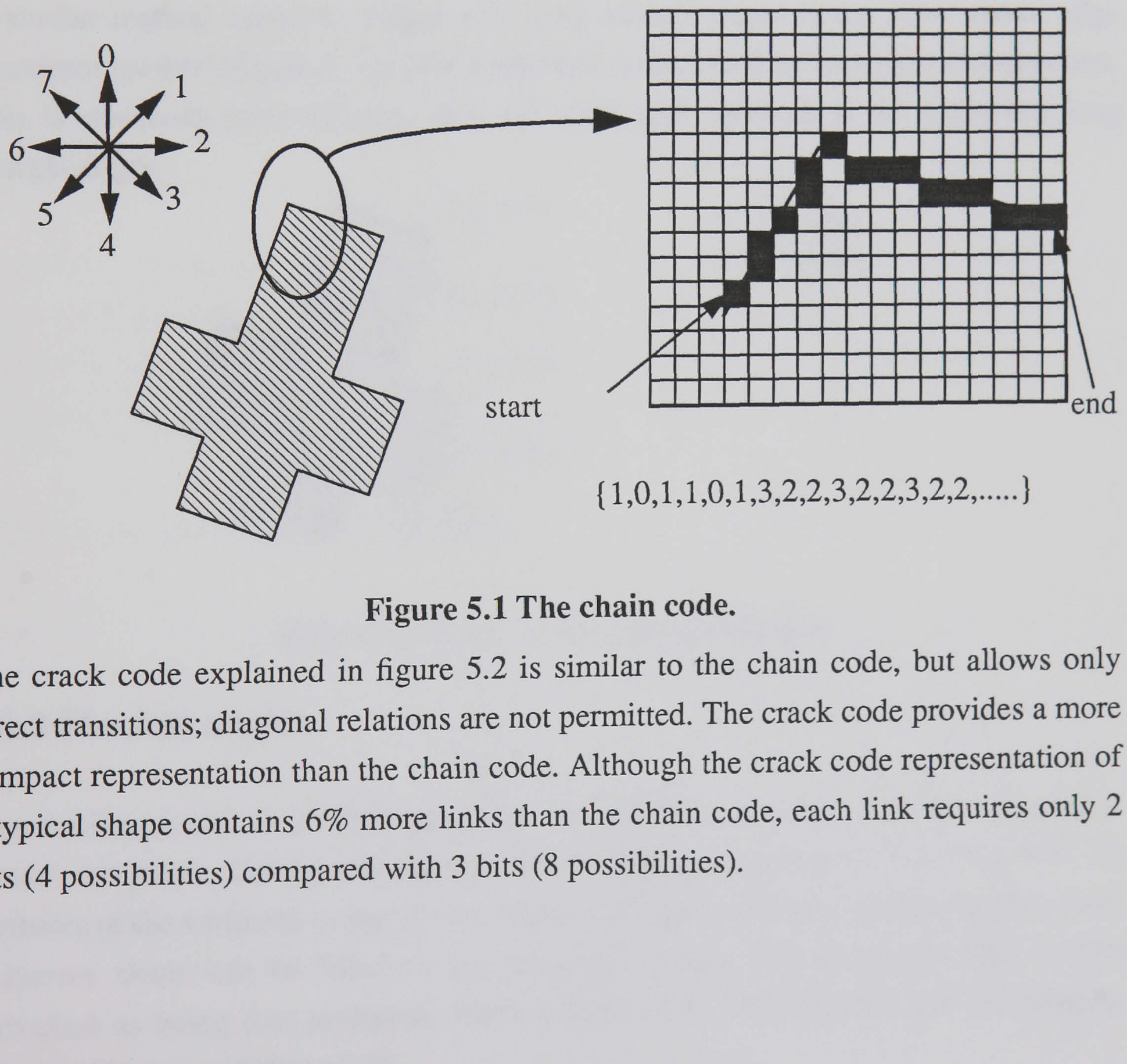


Figure 5.1 The chain code.

The crack code explained in figure 5.2 is similar to the chain code, but allows only direct transitions; diagonal relations are not permitted. The crack code provides a more compact representation than the chain code. Although the crack code representation of a typical shape contains 6% more links than the chain code, each link requires only 2 bits (4 possibilities) compared with 3 bits (8 possibilities).

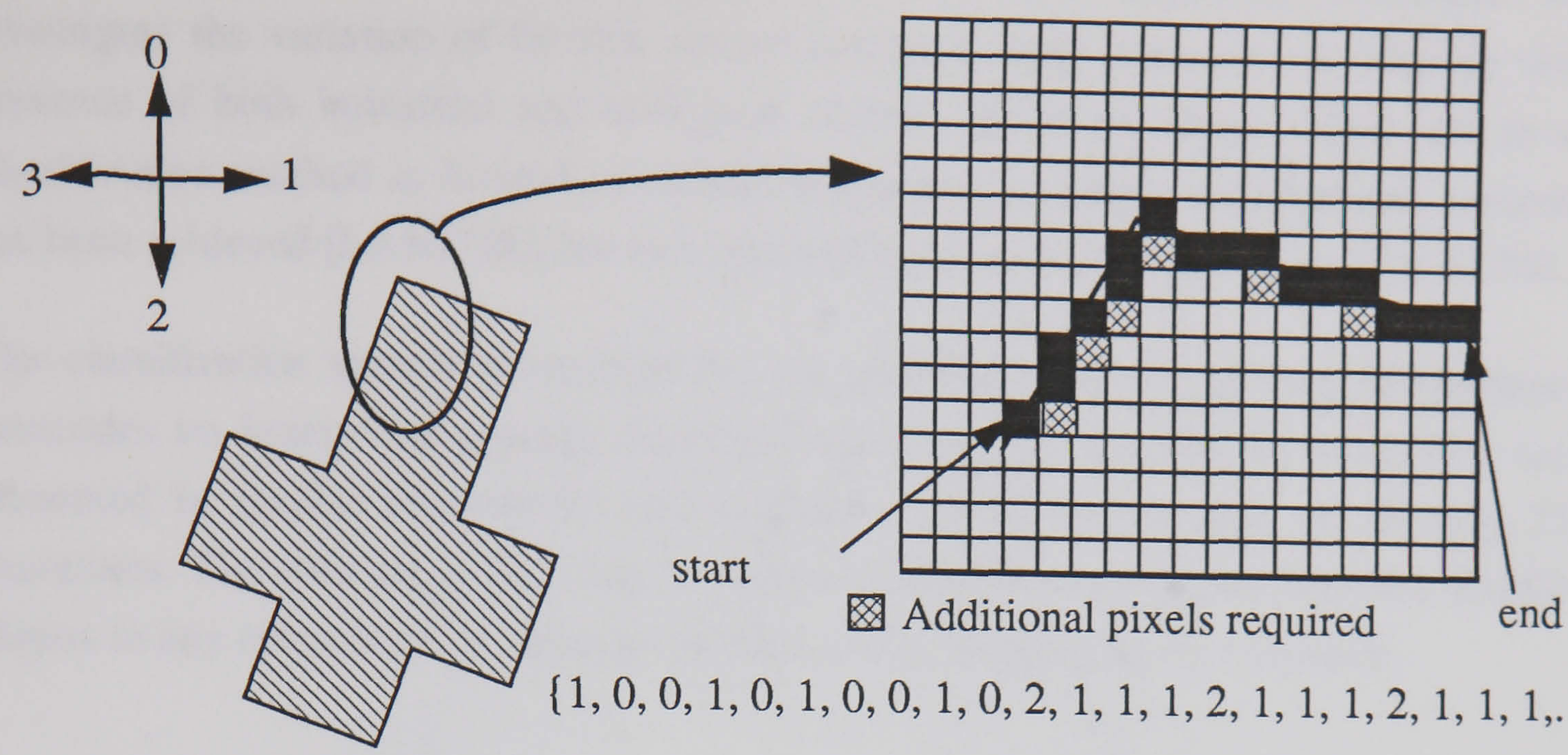


Figure 5.2 Equivalent crack code of outline or contours.

5.1.1.2 Polar vector edge descriptor

A similar method suited to shapes with long straight edges is the polar vector edge descriptor method (figure 5.3) where each edge is described by its length and direction. This is obviously more efficient than the chain code methods if the shape has long straight edges.

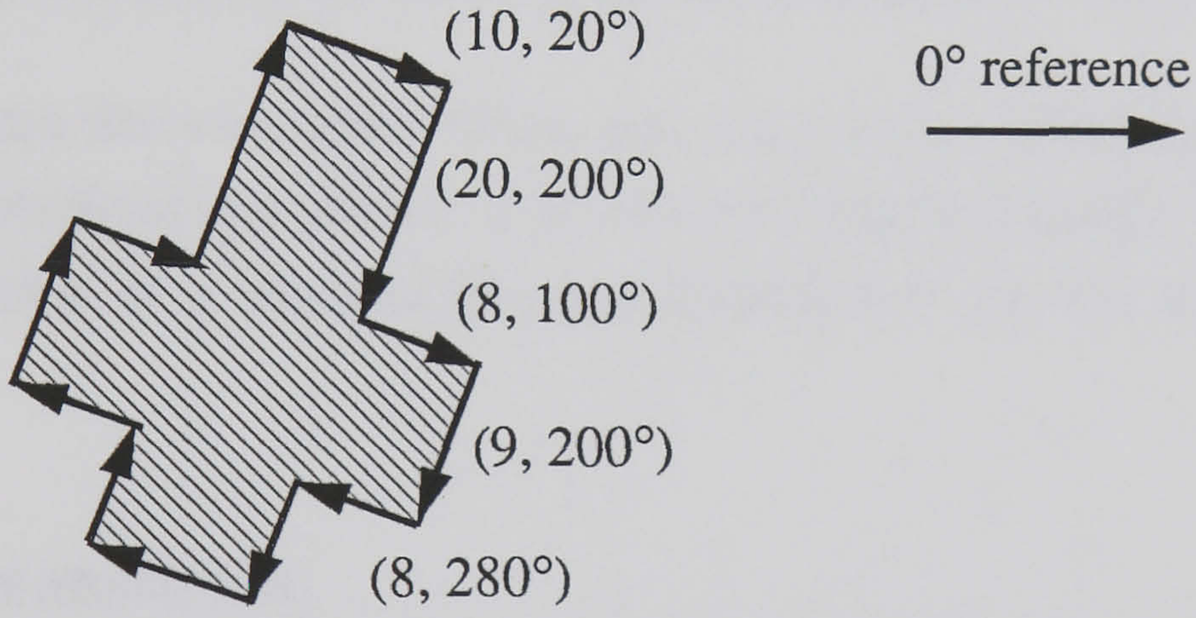


Figure 5.3 Polar vector edge descriptor.

5.1.2 Flexible model

The flexible model of a shape specifies its outline as a series of points distributed around the shape. During training several examples of a shape are used to obtain the statistics of the variation of each point so that during the recognition phase points on an unknown shape can be fitted to the prototype points. The unknown class is then identified as being that prototype which requires the least distortion of its points in order to obtain a satisfactory fit.

Flexible or Point Distribution Models (PDM's) were originally developed to investigate the variation of flexible shapes but have since been used to identify the presence of both industrial and biological objects within an image. Their use as a classification method is limited to character recognition where experimental success has been achieved [LANITIS], but to a lesser extent than would be required by BUSM.

The classification system researched the use of PDM's for recognising handwritten postcodes on letters and parcels. Although handwritten characters do vary, they are presented in similar orientations and originate from a limited data set of only 35 characters. The data set is therefore of a more pre-determined nature than the 20,000 shapes in any orientation, without priori knowledge, targeted by this research.

5.1.3 Localised features

Shoe components are frequently grouped together into sets of very similar patterns differing only in size or some distinguishing mark on the boundary. Components which are identical but need to be processed differently are identified by a set of marks or cuts on the edge of the shape. Evans [EVANS 1993] uses shape segmentation to identify occluded shapes. Although the shoe components in this investigation are not occluded the division of the shape into segments would highlight any small differences between shapes which are nominally the same by allowing an area by area comparison.

In order to obtain the local areas along with the corresponding localised features the shape must be obtained and processed as a two-dimensional image. This is not the most efficient utilisation of a linescan imaging system such as that used for shoe upper recognition.

5.1.4 Space transforms

There are a variety of methods available for image interpretation which work by transforming the image representation from the two-dimensional visual plane to some other domain. This transformation is achieved by the application of a mathematical formula and results in an 'image' often unrecognisable to the user but which contains the rotation, translation and scale invariant measures needed for identification.

5.1.4.1 Two dimensional Fourier transform

The two dimensional Fourier transform [e.g. HUSSAIN 1991] analyses the frequencies of the image, producing Fourier coefficients which represent the relative modulus of

each frequency. The lower order frequencies reflect the global shape characteristics while the high order frequencies depend upon the edge detail of the shape and are more sensitive to localised shape variations. If sets of shapes are very similar then higher order coefficients will have to be calculated in order to achieve separation.

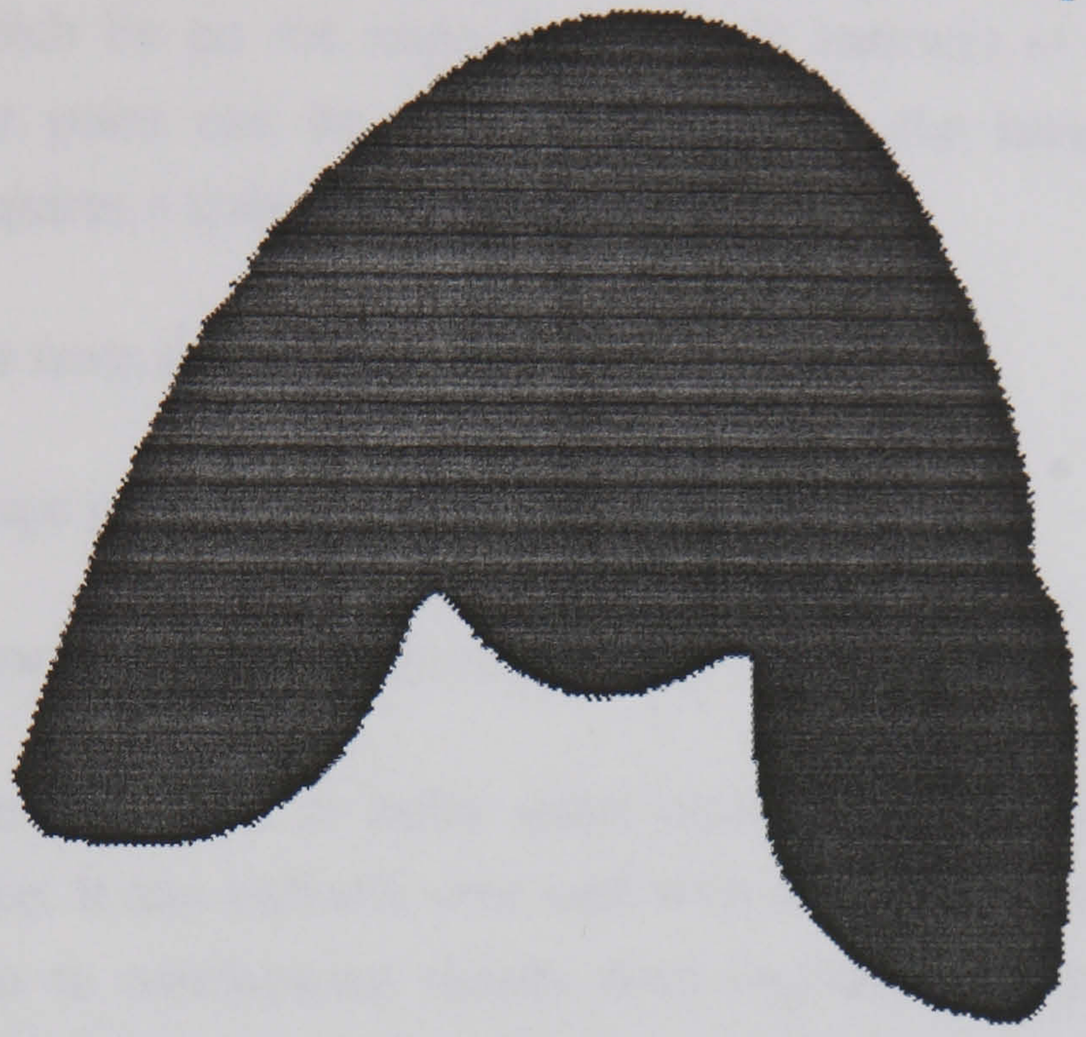


Figure 5.4 An example of a Vamp (Tsh09) component.

Figure 5.4 shows an example of a class in 2D vision space while Figure 5.5 shows the same shape transformed to the Fourier domain. Figure 5.5 reflects the size and phase of the frequency components as determined by the transform.

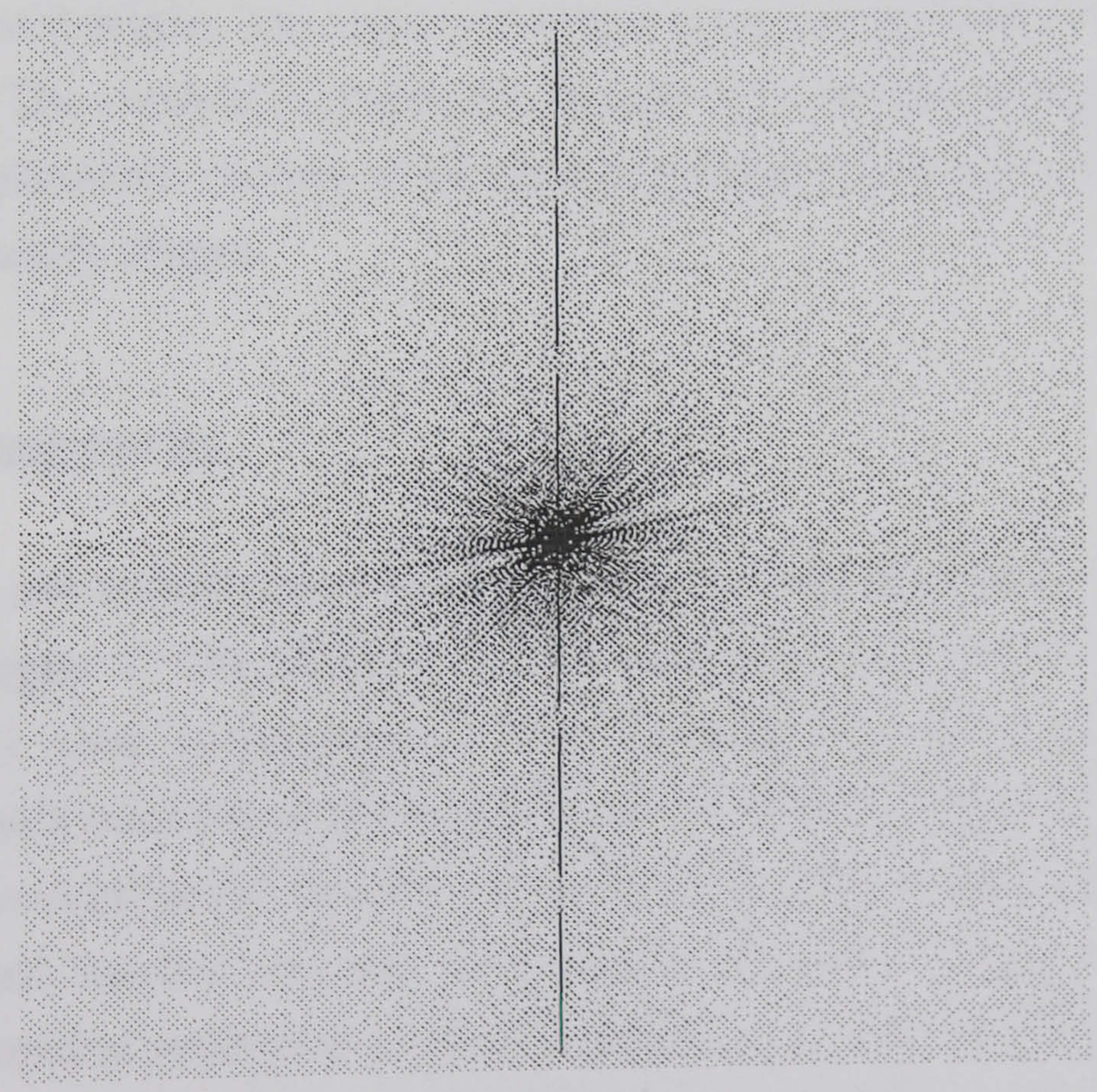


Figure 5.5 The Fourier domain image of component Tsh09.

5.1.4.2 Hough transform

In general the Hough transform [e.g. LEAVERS 1992] plots a series of curves in the Hough plane with each curve corresponding to one edge pixel in the image. The curves of those pixels which lie on the same line will all intersect at some point and the coordinates of that point can be used to characterise the image. To obtain these reference points requires a three stage process:-

- Extract the edges from the image.
- Transform the edge points into the Hough domain.
- Identify the characteristic interception points.

The method is very resilient to noise since noise pixels are unlikely to form the required type of line. It can perform very well with occluded images provided the loss of edge pixels due to overlapping shapes does seriously reduce the density of the overlap point in the Hough space. Unfortunately all of the lines that are to be detected must be parameterised for the method to work correctly. This is achievable for geometric shapes like nuts, bolts and washers but not for shoe components which tend to have 'artistic' edges.

For non-geometric shapes the generalised Hough transform [BALLARD 1981] can be used, this parameterises the shape as a series of vectors connecting each point of the image edge to some origin. The vector elements are the length and direction of the line from the origin. The method is somewhat weaker than the true Hough Transform and requires all the plausible vectors to be tried if arbitrary orientation is involved when attempting to identify shapes.

5.1.5 Neural networks

Neural networks are widely reported in papers on pattern recognition and are reviewed more fully in chapter 7. The ability of neural networks to generalise to the extent that they can classify shoe components is also demonstrated in chapter 7. The major drawback with such methods is the requirement for extended sets of training data. This makes them unfavourable for situations such as shoe upper recognition where training is to be performed 'on-line' as part of the production process and would require the presentation of several examples of each component shape. They have not, however, been completely discarded and as chapter 8 shows they can be successfully employed to identify awkward component groupings without too great a time penalty.

5.1.6 Moments and moment invariants

The $(p,q)^{th}$ moment of a shape F(x,y) about the origin of the coordinate system (x=0, y=0) is given by:

$$m_{p,q} = \int_{x_a y_a}^{x_b y_b} \int F(x,y)\, x^p y^q dx dy \ldots\ldots\ldots\ldots \text{eq } 5.1$$

Which is applied to every pixel within an image in order to extract a series of values based on pixel irradiance and position. The moments themselves are not translation or orientation independent and can be used to determine the alignment and position of objects. Relatively simple combinations of the basic function are invariant with respect to object rotation and translation. Thus they can be used as features to describe the image and compared with values for other images in order to perform identification.

In theory the higher the order of the moment the more responsive it will be to detailed changes in component shape as occur at the perimeter. In practice the perimeter of a shape tends to be noisy and this fact combined with the larger powers of x and y used to calculate the higher order moments means that moment calculations tend to be susceptible to noise. The minimal amount of pre and post processing needed makes moments an attractive proposition for industrial vision tasks and they have been used for the recognition of characters [HU 1961], ships [SMITH *et al.* 1971] and aircraft [DUDANI *et al.* 1977].

Blumenkrans [BLUMENKRANS 1991] made direct experimental comparisons between a polar recognition method and a moment recognition method. He concluded that the moment method was superior for the identification of whole silhouette shapes while the polar method performed better with line images. The shapes used were considerably different to one another compared to some shoe component examples.

5.2 Conclusions

Tout [TOUT 1989] continuing the work of Koulopoulos maintained that area and radii descriptors were the best method of representing shoe shapes. They were easy to implement and had proved successful during many tests. The method for finding shape orientation from radii relies on the comparison of 360 radii in each possible shape position. This is not an economical or efficient method. In addition, a different set of features are needed in order to locate the centroid of a shape. The ideal representation

of a shape should be able to perform all three tasks (identification, centroid location and shape orientation) efficiently and economically from the same basis set of features.

Moment based features offer the opportunity to represent shapes in an efficient manner and, as chapter 6 will show, this efficiency extends to the computation and general handling of the moments in a manner which is compatible with the compressed data format.

The proposed use of moment based methods contradicts the work of Koulopoulos who rejected their use on the grounds that they lacked a straight forward interpretation. It would seam that Koulopoulos severely limited his choice by using this criteria in the selection of features and a powerful recognition technique remained largely unevaluated due to this restriction. Koulopoulos also considered moments too cumbersome for efficient processing, a problem which has been overcome by calculating moments based on the compressed image.

Tout did not evaluate the use of moments beyond the inclusion of the second-order moments for orientation and as additional features in his feature set. Tout was, however, concerned primarily with research into the whole process of automatic stitching and did not dwell on the vision system, preferring to simply review and modify the work of Koulopoulos.

Chapter 6 describes in more detail the work done with moments and moment invariants.

CHAPTER 6

MOMENTS AND INVARIANT MOMENTS

6.1 Describing a shape with moments

Consider a continuous two dimensional image with respect to some origin. This image can be described by considering functions (moments) based on the irradiance and position of each image point. In general, the moments about the origin for an image within an area x_a to x_b by y_a to y_b are described by [HU 1961]:-

$$m_{p,q} = \int_{x_a}^{x_b}\int_{y_a}^{y_b} F(x,y)\, x^p y^q dx dy \ldots\ldots\ldots\ldots \text{eq } 6.1$$

for p=0, 1, 2.....∞; and q=0, 1, 2.....∞;

where

$m_{p,q}$ is a moment of $(p + q)^{th}$order.

x_a, x_b, y_a, y_b are the limits along the x and y axis of the image.

F(x,y) is the irradiance function, i.e. brightness of the image at the point (x, y).

An image is completely described by the set of all possible moments;

$m_{0,0}$, $m_{0,1}$, $m_{1,0}$, $m_{2,0}$, $m_{1,1}$, $m_{0,2}$, $m_{3,0}$, $m_{2,1}$,.....$m_{\infty,\infty}$

In practice only a finite number of moments need to be calculated in order to represent an image to a given accuracy. The more moments used, the better the approximation to the true image (this is similar to the reconstruction of signals from Fourier components).

For calculation by computer the discrete form of the equation must be used;

$$M_{p,q} = \sum_{x=x_a}^{x_b} \sum_{y=y_a}^{y_b} F(x,y)\, x^p y^q \ldots\ldots\ldots\ldots \text{eq } 6.2$$

The values x and y now take on integer values only and refer to pixels within the image. The value $M_{p,q}$ is a finite value between the limits of the image, x_a to x_b and y_a to y_b. F(x, y) refers to the brightness of the $(x, y)^{th}$ pixel.

6.2 The shape centroid and translation normalised moments

Typically x and y will be referenced to the image origin. Any movement (translation) of a component within the image field will produce changes to $M_{p,q}$ values. To overcome this x and y are referenced to the centroid of the shape to produce normalised moments. The centroid (X, Y) of a two dimensional binary (silhouette) shape can be found using;

$$X = \frac{M_{10}}{M_{00}} \ldots\ldots\ldots\ldots \text{eq } 6.3$$

$$Y = \frac{M_{01}}{M_{00}} \ldots\ldots\ldots\ldots \text{eq } 6.4$$

where F(x,y) = 1 for pixels within the shape.

F(x,y) = 0 otherwise.

Having found the coordinates of the centroid the translation normalised moments (denoted by μ) can be calculated;

$$\mu_{p,q} = \sum_{x=x_a-X}^{x_b-X} \sum_{y=Y-y_a}^{Y-y_b} F(x,y)\, x^p y^q \ldots\ldots\ldots\ldots \text{eq } 6.5$$

These values will remain constant as the component is moved throughout the image field provided that its orientation is not disturbed.

6.3 Orientation (and rotation) normalised moments

Hu [HU 1961] discusses the use of image moments as the basis for an invariant set of features to be used for shape recognition and presents a method of combining transition invariant moments to form orientation independent groups. Although Hu's work is based on that of nineteenth century mathematicians (Boole, Cayley and Sylvester) it is the first significant attempt to use high order moment invariants for two dimensional image recognition.

Some combinations of transitional invariant moments which are also invariant to rotation are: (note: for clarity $\mu_{p,q}$ etc. is replaced by μ_{pq} where $p \leq 9$ & $q \leq 9$,)

$$F_0 = \mu_{00}$$

$$F_1 = \mu_{02} + \mu_{20}$$

$$F_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$F_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$

$$F_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

$$F_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2)$$

$$F_6 = (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$$

$$F_7 = (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$$

$$+ (4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}))$$

The derivation and method for calculating these moment invariants can be found in Hu's paper on 'Visual Pattern Recognition by Moment Invariants' [HU 1961].

6.3.1 Zernike moments

Teague in his paper on 'Image Analysis Via the General Theory of Moments' [TEAGUE 1980] presents an alternative method for obtaining sets of invariant moments based on the use of Zernike polynomials. Teague's method produces sets of invariants which are slightly different to those of Hu and considerably easier to generate when moments of a higher order are required. The general method, however, remains the same.

6.4 The feasibility of moments as features

There are several issues which need to be considered when evaluating the usefulness of moment invariants as features:

- Ability of the feature set to discriminate between shapes.
- Usefulness in determining position and orientation of shapes.
- Ease of calculation and associated speed of processing.
- Size of feature based representation.

In order to establish the feasibility of these measures as features an experiment was performed in which various images of the shapes shown in figure 6.1 were scanned. The feature sets produced for each component could then be used as a basis from which to evaluate moment invariant feature feasibility.

The research system used to obtain these images differed slightly from the Autostitcher machine described in chapter 2. The resolution, illumination and camera position were identical but the gritted rollers for moving the component had to be replaced by a moving sheet of glass onto which the component was placed. The research system used BUSM's latest vision module based on the transputer and DSP.

The results show that moment based features can be used to evaluate all the required information; position, orientation and identity. The moments also have the advantage over many other methods that they can be calculated from the compressed data as each line of the shape is scanned, avoiding the need to store and process whole sections of image.

Tsh11

Tsh12

Tsh14

Tsh15

Figure 6.1 The initial test shapes.

6.4.1 Identification features

Tout established that area is a very stable measure of a component's identity, thus the trial set of shapes all had equal area. Thirty six images were gathered for each shape at 10 degree rotations and thirty six feature vectors were calculated and compared on a feature for feature basis.

Figures 6.2.a to 6.2.d. show the value of each feature for each of the samples taken. Note that although the combinations of moments are theoretically invariant the value of the features varies with the changing orientation of the shape due to the quantisation error of the binary system.

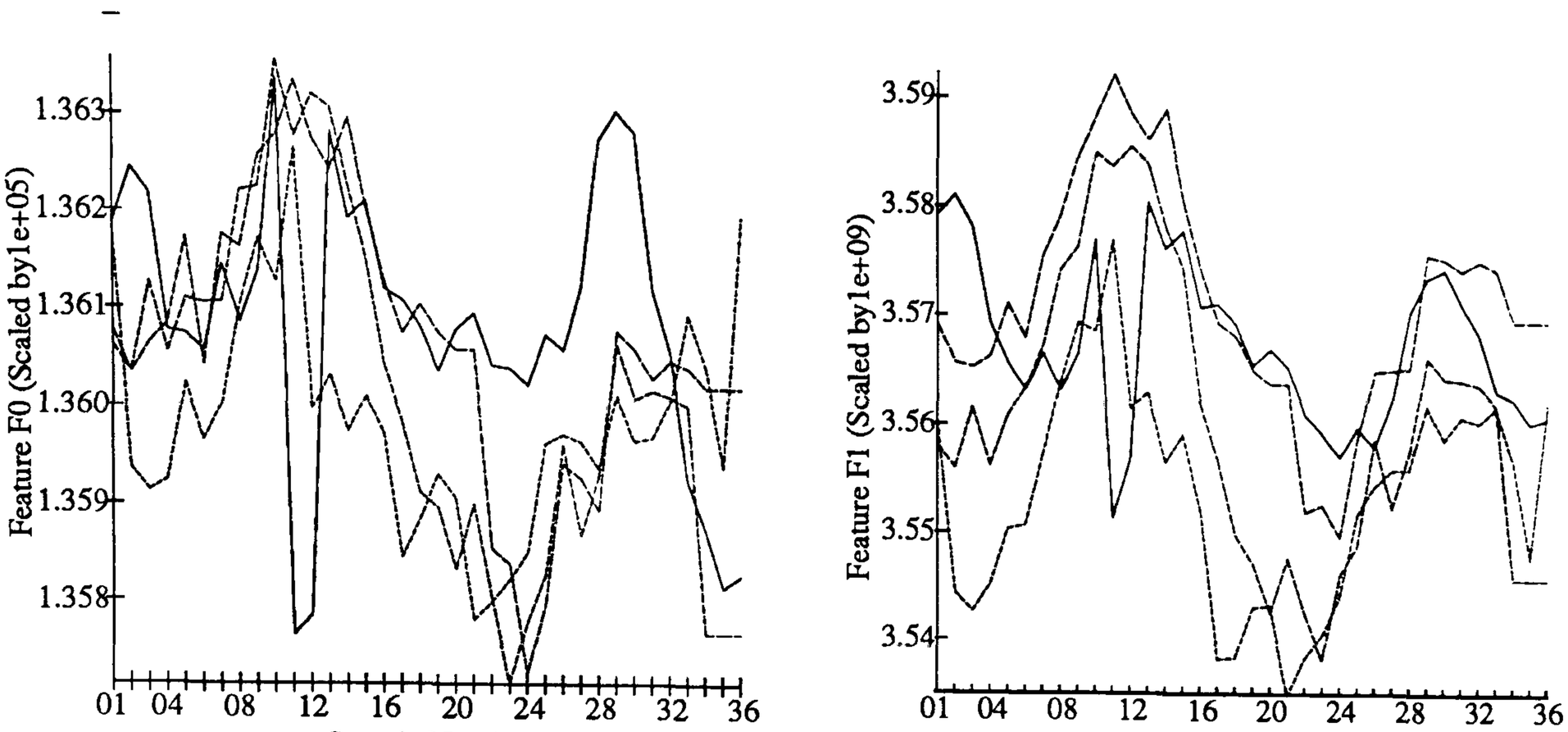


Figure 6.2.a Invariant features F0 and F1 for 36 samples at 10° intervals.

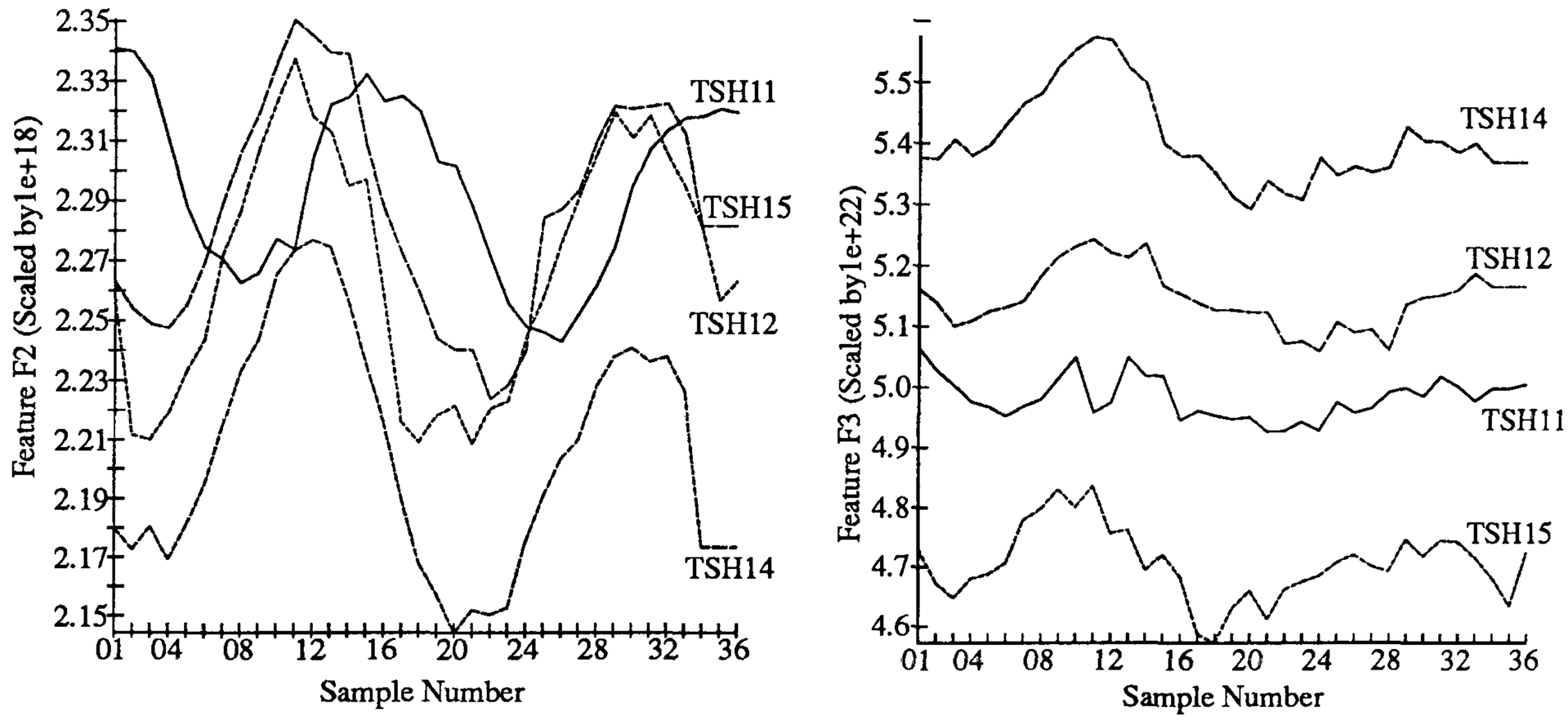


Figure 6.2.b Invariant features F2 and F3 for 36 samples at 10° intervals.

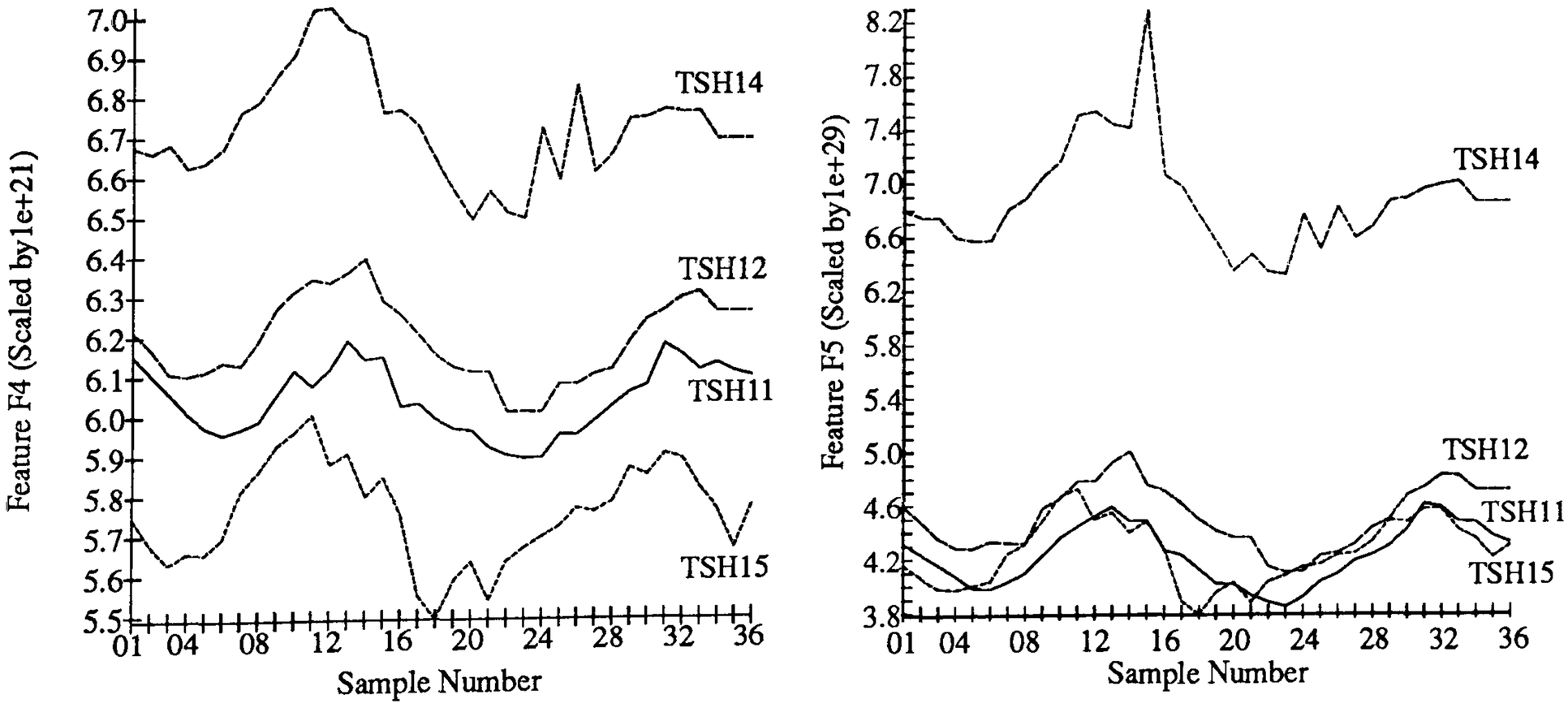


Figure 6.2.c Invariant features F4 and F5 for 36 samples at 10° intervals.

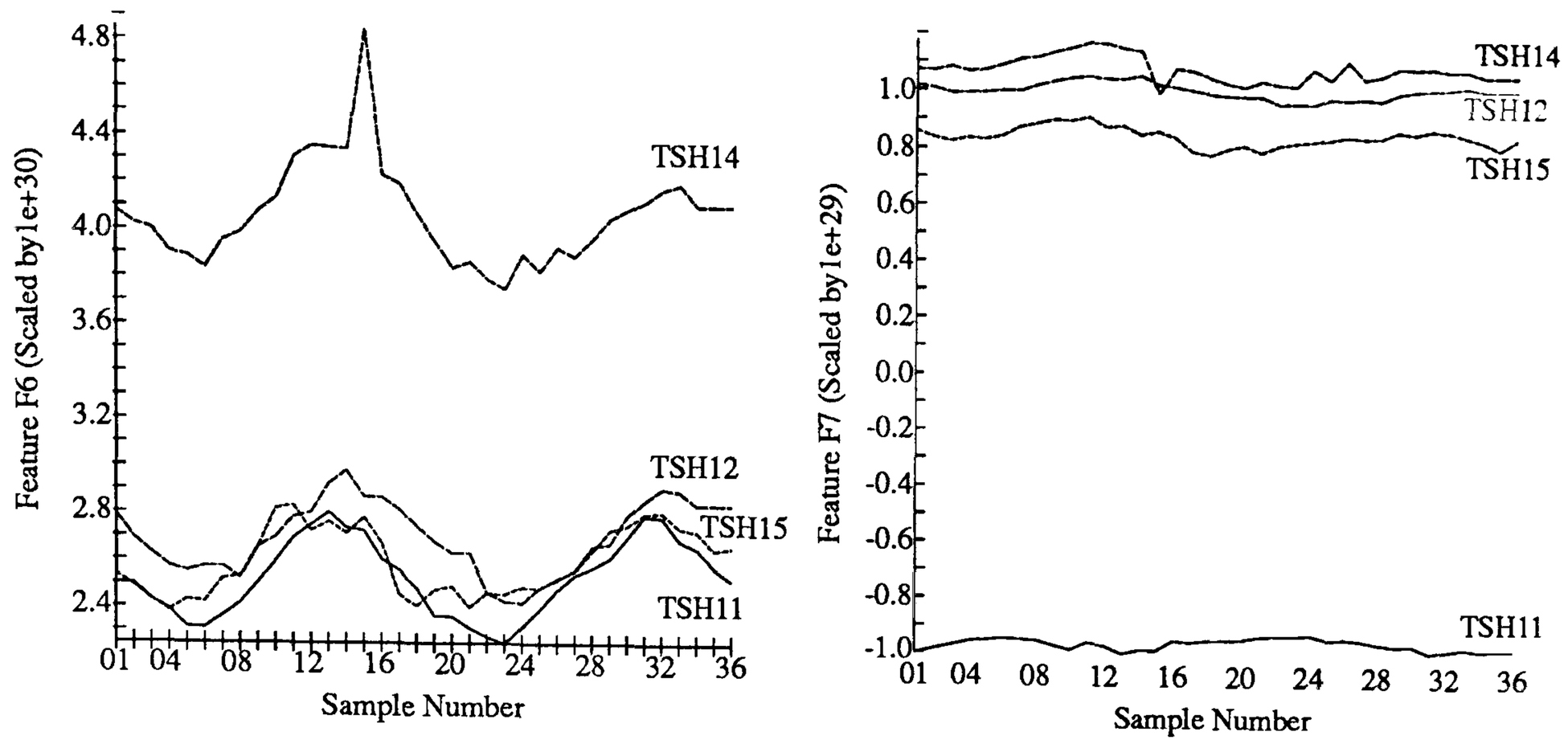


Figure 6.2.d Invariant features F6 and F7 for 36 samples at 10° intervals.

In the case of this group of components (figure 6.1) it can be seen that the lower order moments (F0 to F2) are inseparable due to the similarity of the four components. The lower order moments reflect the basic shape characteristics while the higher order moments reflect progressively more shape detail. The five higher order moments (F3 to F7) show differences in value and can be used to separate the shapes.

6.4.1.1 Closeness of shapes

In order to quantify a systems ability to distinguish shapes, it is useful to determine some measure of the similarity between shapes.

If two silhouette shapes are placed one over the other then some pixels will have a common value while others will differ between the shapes. If the shapes are moved and rotated the numbers of common and different value pixels will change. At some point the number of pixels with a common value will be a maximum and this value can be used to evaluate shape similarity by using the relationship defined in equation 6.6.

$$\text{SIMILARITY} = \frac{\text{PS}}{PD + \text{PS}} \times 100\% \Big|_{\text{PS}_{max}} \quad \text{.....eq 6.6}$$

where:

PS = pixels set to '1' in both components.

PD = pixels set to '1' in component A and '0' in component B or vice versa.

measured when PS is maximum.

Any two components in Figure 6.1 are 98.9% similar.

6.4.1.2 Extending the initial experiment

As shapes become more and more similar then they obviously become harder to separate. In initial tests shapes which were less then 99% similar showed good separation while those of 99% and above produced results suggesting they would prove harder to identify. Figure 6.3 shows components Tsh09 and Tsh16 which are 99.8% similar. They are, in fact, identical components with coding 'notches' for different operations. Successful separation of these two shapes averaged only 75% in the early experiments. Methods of increasing the separation between such 'difficult' shapes are presented later in this chapter along with a larger and more analytical set of identification results for the moments-based system.

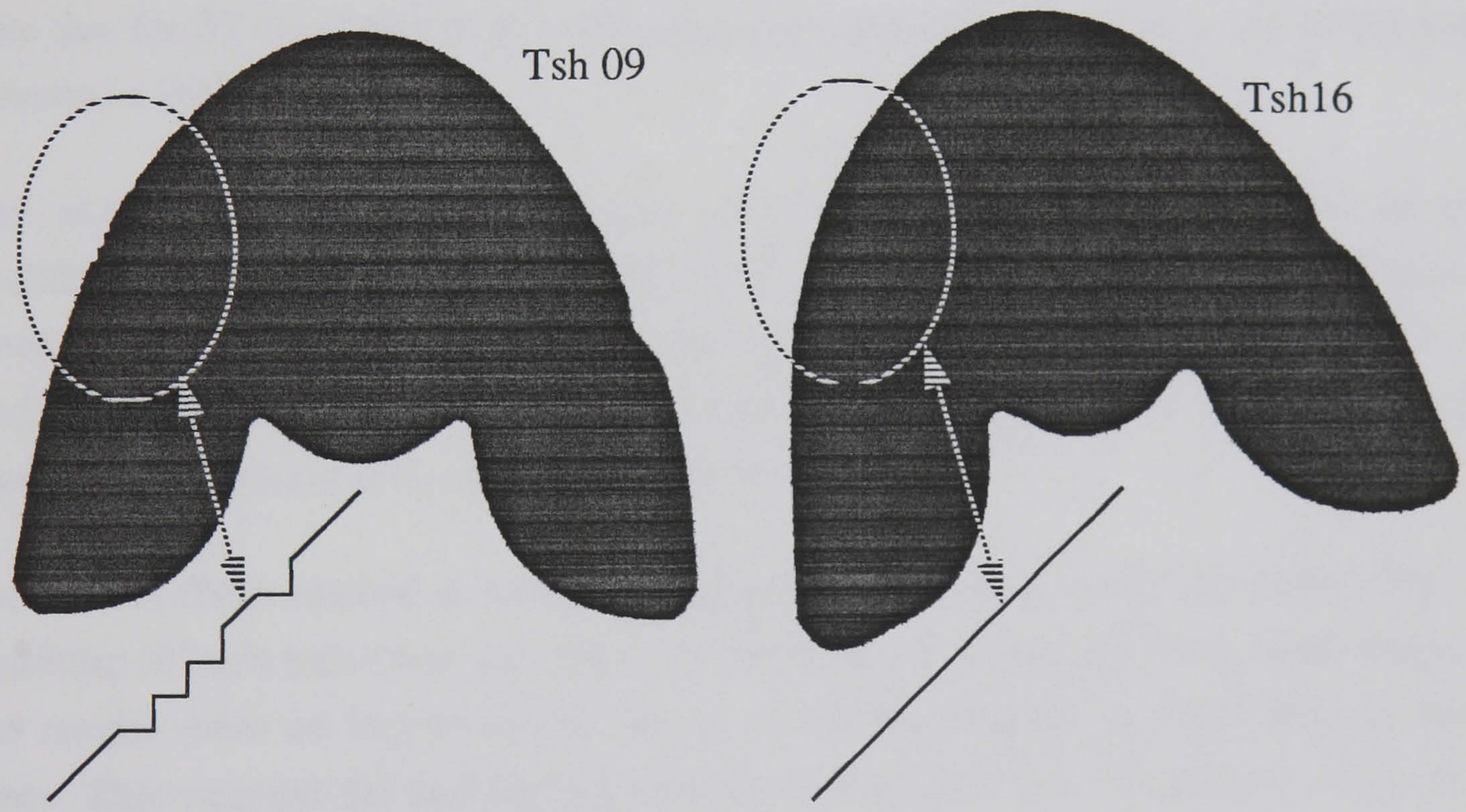


Figure 6.3 Components Tsh09 and Tsh16 are 99.8% similar.

6.4.1.3 Experimental system setup

The following section describes the set up for the system used to evaluate moment invariant based component identification. It applies to the investigations covered in sections 6.4.1.4, 6.6.1.1 and 6.6.2.1.

The primary pattern recognition was performed using the feature matching method discussed in section 4.3, the tolerances used were:

Feature	Tolerance (%)
F0	0.3
F1	1.5
F2	3.0
F3	4.0
F4	6.0
F5	20.0
F6	30.0
F7	same sign
Minimum Match	5 features from 8

Table 6.1 Experimental tolerances.

Note that for F7 the shape value need only match the prototype sign (+ve / -ve) in order to record a feature match.

Any shapes matching more than one prototype were initially identified using a Euclidean distance measure [e.g. BROWNE 1986] to assess the distance in feature space between the shape and each prototype. This gave poor results because the Euclidean distance measure was dominated by the magnitude of the highest order feature. To overcome this, two methods were tried:

The first of these applied a nonlinear scaling to the feature vector by calculating the logarithm of each value and then found the Euclidean measure for these 'new' features. The results were an improvement, but were still dominated by single feature value errors. This negated the feature redundancy incorporated into the feature comparison based on the Hamming code principle.

The second method, which is used in the tests, simply compares the unknown shape with the possible prototypes on a feature by feature basis, effectively calculating a one dimensional Euclidean distance. The prototype with the smallest Euclidean distance for that feature 'scores' one point and the component is assigned to the prototype with the highest score. If two prototypes score equal highest points then the component is assigned to the first prototype.

The actual machine used was the research system as described in section 6.4.

6.4.1.4 The use of moment invariants for identification

Table 6.2 shows the average results of the earliest experiments when moment invariant features were used to distinguish a test set of 20 components. The components were picked so as to present the system with several sets of components from the same family. The components used and their groups are given in appendix IV.

TRUE SHAPE \ ALLOCATED SHAPE	Tsh01	Tsh02	Tsh03	Tsh05	Tsh06	Tsh08	Tsh09	Tsh10	Tsh11	Tsh12	Tsh13	Tsh14	Tsh15	Tsh16	Tsh19	Tsh20	Tsh21	Tsh23	Tsh28	Tsh38	REJECT
Tsh01	90.6																				9.4
Tsh02		100																			
Tsh03			89.4																10.6		
Tsh05				100																	
Tsh06					99.4																0.6
Tsh08						94.4															5.6
Tsh09							74.4							25.6							
Tsh10								100													
Tsh11									85.5	12.8	0.6		1.1								
Tsh12									5.6	93.8			0.6								
Tsh13									2.5		97.5										
Tsh14										4.4		92.3									3.3
Tsh15									1.7				98.3								
Tsh16							25.7	0.4						69.5	2.2						2.2
Tsh19															100						
Tsh20																97.2					2.8
Tsh21																14.4	71.7	12.2			1.7
Tsh23																1.7	1.7	96.6			
Tsh28			5.0																89.4		5.6
Tsh38																				100	

Table 6.2 The results of using moment invariant features F0 to F7.

These results can be summarised as:-

The average substitution rate (components allocated to wrong class) is 6.44%.

The average rejection rate (components unidentified) is 1.56%.

Average total failure rate (sum of substitution and rejection errors) is 8.00%.

The summary results are obtained by adding all the (e.g. substitutions) together, dividing by 2000 (the total number of tests) and multiplying by 100%.

As expected, substitution errors only occur between shapes of the same group (see appendix IV) which leads to the symmetrical nature of the results in table 6.2. On analysis it is found that in substitution cases the correct prototype is often proposed as a candidate by the feature matching process but rejected by the Euclidean stage. The system could be set up so that any component which matches two or more prototypes is rejected for re-scanning. This would increase the level of rejection but decrease the number of more damaging substitutions.

6.4.2 Position locators

The use of equations 6.3 and 6.4 to satisfactorily locate the centroid of a component has been demonstrated by Tout.

The pattern to be stitched is stored relative to the centroid position so it is important that a high degree of accuracy is attained. Tests by Tout showed that by using equations 6.3 and 6.4 the sewing needle could be repeatedly placed into the positions of the original taught stitch marks.

6.4.3 Defining shape orientation

Once a component has been identified the appropriate pattern for stitching can be selected. In order for the operation to be successful the pattern must be rotated to account for the difference in orientation between the prototype shape used to store the pattern and the component about to be stitched.

The simplest method of determining the orientation of a shape is to use equation 3.6 based on the angle of the equivalent ellipse as described in 3.5.3.1.

6.4.3.1 Component condition

As the equivalent ellipse approaches circularity then the quantity μ_{20} - μ_{02} becomes unstable due to the similarity between the two values. The use of equation 3.6 for determining the orientation becomes unreliable due to the effects of any noise in the image (the generation of quantisation noise and noise due to the manufacturing process are discussed in sections 2.2.4 and 3.3 respectively). The actual component being viewed can be completely uncircular as seen in figure 6.4 and still cause this problem. BUSM defined equation 6.7 in order to evaluate the reliability of a component's orientation as determined by equation 3.6.

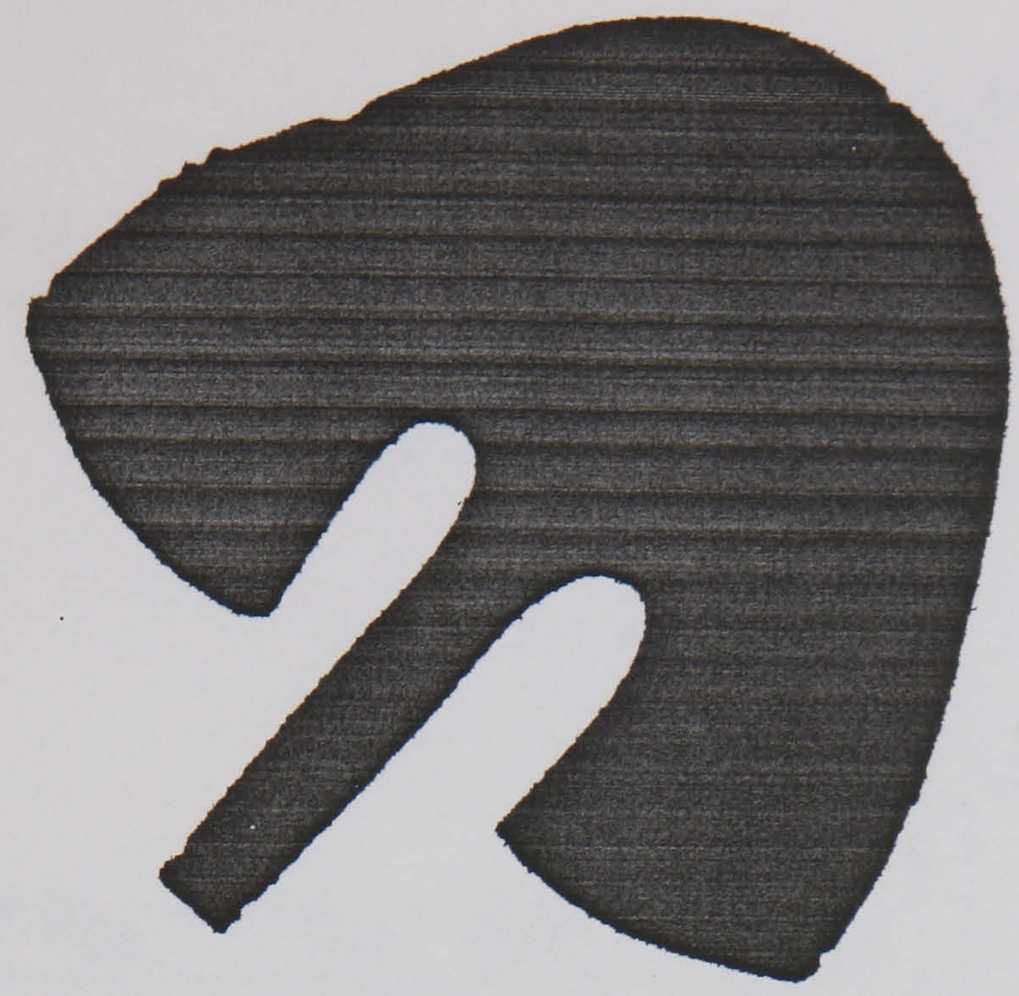


Figure 6.4 An uncircular shape which is easily orientable by human eye.

$$GCOEFF = \frac{\hat{\mu}_{02} - \hat{\mu}_{20}}{\hat{\mu}_{02} + \hat{\mu}_{20}} \ldots\ldots\ldots\ldots \text{eq } 6.7$$

BUSM have determined experimentally [BLOWER 1982] that the critical value for GCOEFF is 0.2. Equation 3.6 will not work satisfactorily for shapes whose GCOEFF falls below 0.2, due to the equivalent ellipse approaching circularity (a perfect circle will have a GCOEFF value of 0.0) and the increased distortion caused by noise pixels on the calculation. An alternative method must be used for these shapes

GCOEFF is evaluated by rotating the shape (in software) to find the maximum values of μ_{20} and μ_{02}. When μ_{20} and μ_{02} are maximum values μ_{11} will be zero and the coordinate axes will be aligned with the principal axes of the shape.

This method of determining GCOEFF is computationally expensive. This investigation has devised a less intensive version which relies on directly calculating the major and minor axis of the equivalent ellipse without the need to align the principal axis of the shape with the coordinate system.

The major (A) and minor (B) axes of the equivalent ellipse can be calculated from:

$$A = \left(\frac{\mu_{20} + \mu_{02} + ((\mu_{20} - \mu_{02})^2 + 4\mu^2_{11})^{\frac{1}{2}}}{\frac{\mu_{00}}{2}} \right)^{1/2} \quad \text{.....eq 6.8}$$

[TEAGUE 1980]

$$B = \left(\frac{\mu_{20} + \mu_{02} - ((\mu_{20} - \mu_{02})^2 + 4\mu^2_{11})^{\frac{1}{2}}}{\frac{\mu_{00}}{2}} \right)^{1/2} \quad \text{.....eq 6.9}$$

[TEAGUE 1980]

The value of GCOEFF can then be calculated as

$$GCOEFF = \frac{A^2 - B^2}{A^2 + B^2} \text{.....eq 6.10}$$

6.4.3.2 Use of higher order moments

The method using equation 3.6 is computationally efficient but not reliable in those cases where GCOEFF is less than 0.2. The alternative method used by Tout and described in 3.5.3.2. is inefficient and cumbersome but does provide the required accuracy.

This work has shown that an alternative method based on the use of higher order moments is reliable enough for those occasions when the component is ill conditioned (GCOEFF is less than 0.2);

For a given component rotated from x to x' the moment μ_{30} will become μ_{30}':-

$$\mu'_{30} = \sum_{\text{all } y}\sum_{\text{all } x} x'^3 \ldots\ldots\ldots\ldots \text{eq } 6.11$$

x' can be calculated in terms of the original coordinate values (x & y) prior to rotation;

$$x' = x\cos\theta + y\sin\theta \ldots\ldots\ldots \text{eq } 6.12$$

therefore:

$$\mu'_{30} = \sum_{\text{all } y}\sum_{\text{all } x} (x\cos\theta + y\sin\theta)^3 \ldots\ldots\ldots\ldots \text{eq } 6.13$$

which gives:

$$\mu'_{30} = \sum_{\text{all } y}\sum_{\text{all } x} x^3\cos^3\theta + y^3\sin^3\theta$$

$$+3x^2y\cos^2\theta\sin\theta + 3xy^2\cos\theta\sin^2\theta \ldots\ldots\ldots \text{eq } 6.14$$

prior to any change in orientation:

$$\sum_{\text{all } y}\sum_{\text{all } x} x^3 = \mu_{30} \ldots\ldots\ldots\ldots\ldots\ldots \text{eq } 6.15$$

therefore:

$$\mu'_{30} = \mu_{30}\cos^3\theta + \mu_{03}\sin^3\theta + 3\mu_{21}\cos^2\theta\sin\theta$$

$$+3\mu_{12}\cos\theta\sin^2\theta \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{eq } 6.16$$

If the prototype had moments μ_{00}, μ_{01}, μ_{10},..... μ_{30} and the new component with the same identity has moments μ'_{00}, μ'_{01}, μ'_{10},.... μ'_{30} then equation 6.16 can be solved iteratively (e.g. Newton-Raphson interpolation) to find the value of theta required to rotate the moments of the prototype shape to give the μ'_{30} value found in the new component.

In order to evaluate this method as compared to that using the equivalent ellipse a simulation experiment was performed. Component Tsh03, which has a GCOEFF value of 0.088, was scanned. The features were then rotated through 9.0° using software based on equation 6.12 and the y equivalent shown in equation 6.17.

$$y' = -x\sin\theta + y\cos\theta \quad \text{.....eq 6.17}$$

Various amounts of random noise were added to these rotated features and the orientation then re-calculated based on equation 3.6 and the iterative solution to Equation 6.16 (after 10 passes). Figures 6.5 and 6.6 show the range of deviation from 9.0° compared with the maximum noise added for each method.

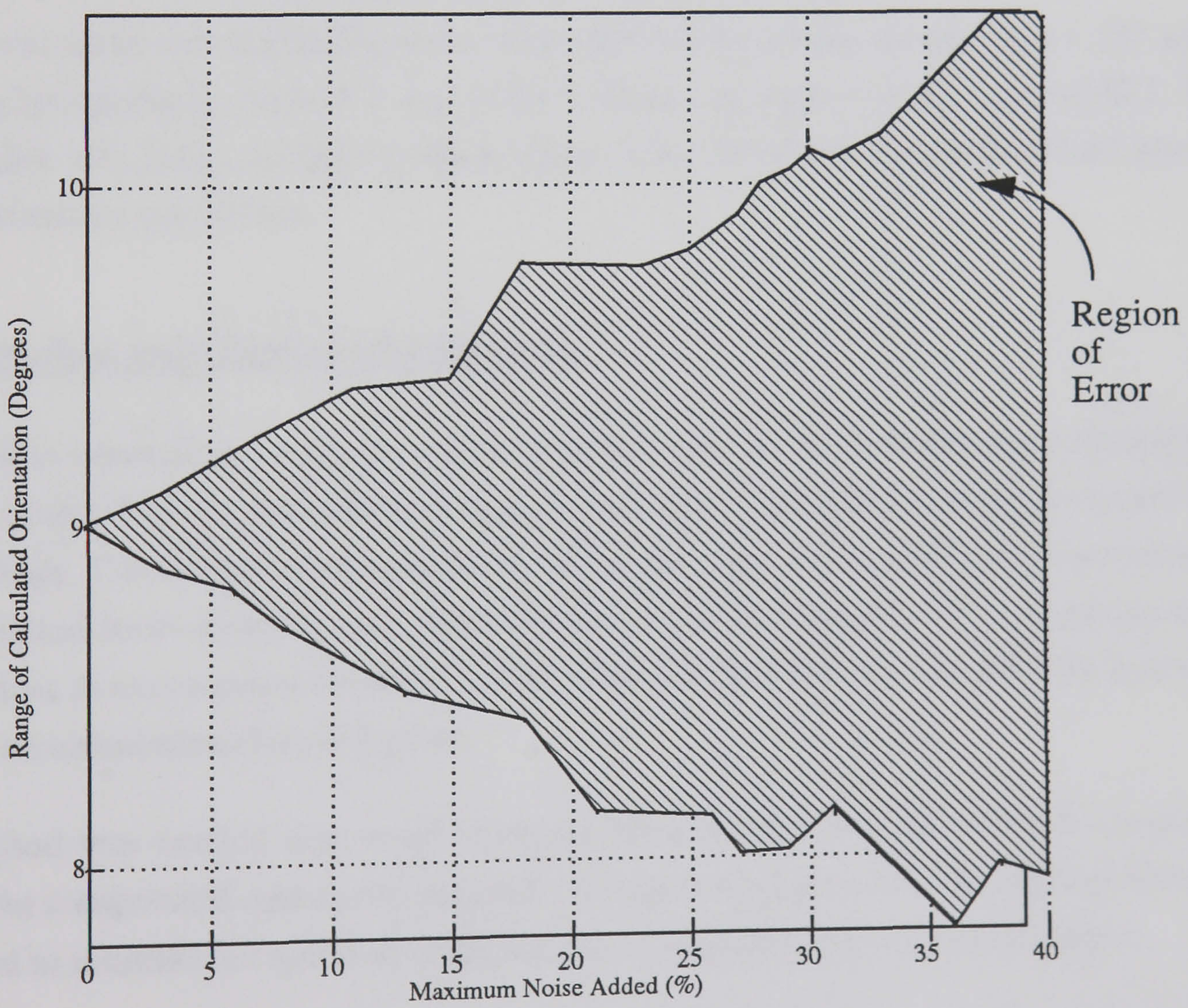


Figure 6.5 Error due to use of equation 3.6.

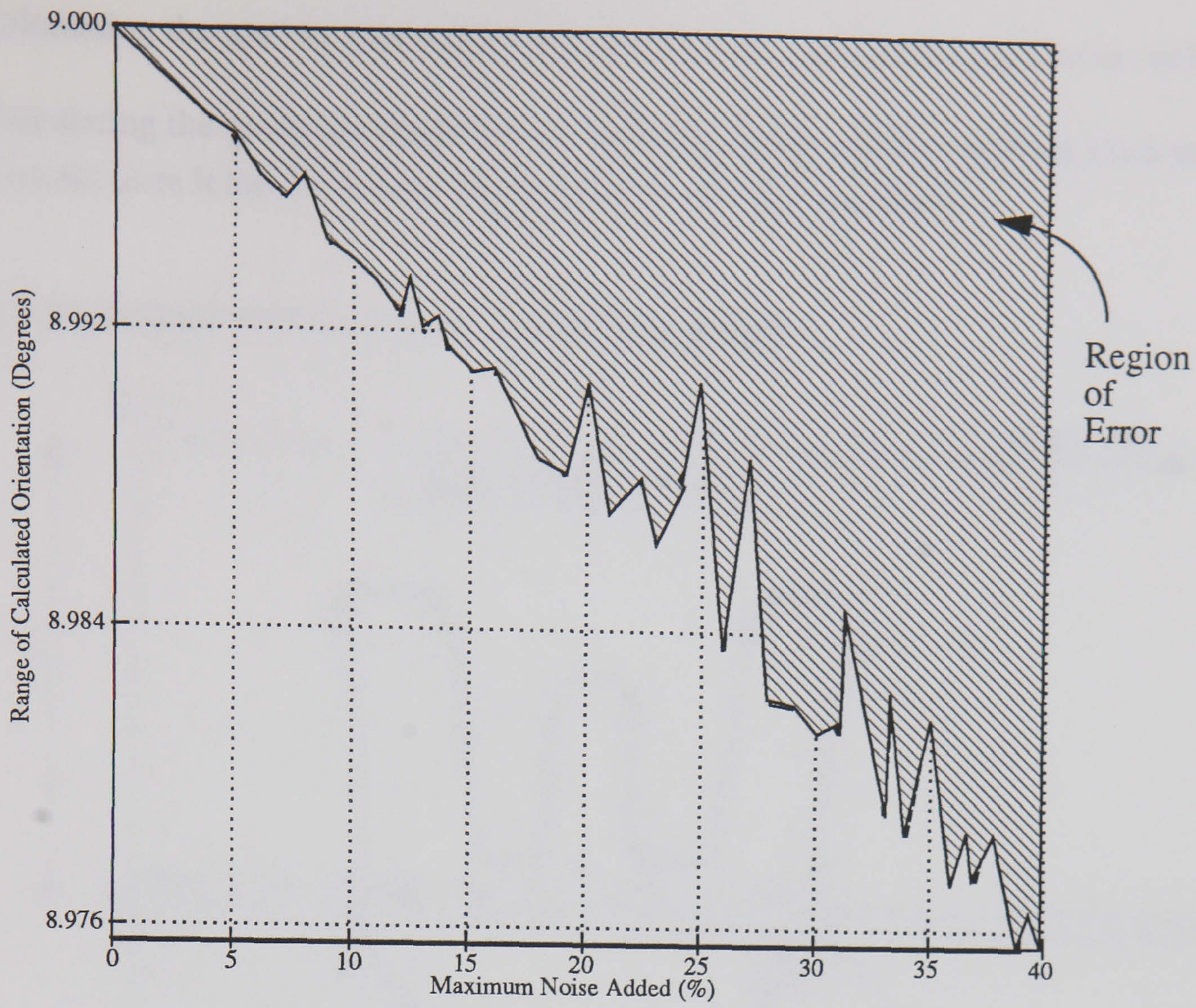


Figure 6.6 Error due to Iterative Method

When the noise was added the worst case error for the ellipse method was 1.23° while for the Interpolation method it was 0.025°. This is an improvement ratio of 49.2. The drawback with such a method is that it relies on the component being identified prior to the orientation calculation.

6.5 On-line calculation of moments

Transition normalised moments which form the basis of the feature set are normalised to the centroid of the component image and based on calculations using every pixel in that image. The images are transmitted to the signal processors for feature extraction in compressed form detailing only the addresses of pixel changes. It is computationally expensive to uncompress the entire image, find the centroid and then find the moments based on calculations for each pixel.

A method was needed that could calculate the moments and the moment invariants from the compressed data as it is scanned. The following describes the method that was derived to increase the speed of computation; it comprises the following stages:-

- Extracting the moments of the image on a line by line basis as they are scanned by the camera.

- Calculating the total moments for each line from the compressed data values only.
- Calculating the moments relative to an arbitrary origin and then shifting them to the centroid once it has been determined when scanning is complete.

6.5.1 Moments relative to an arbitrary origin

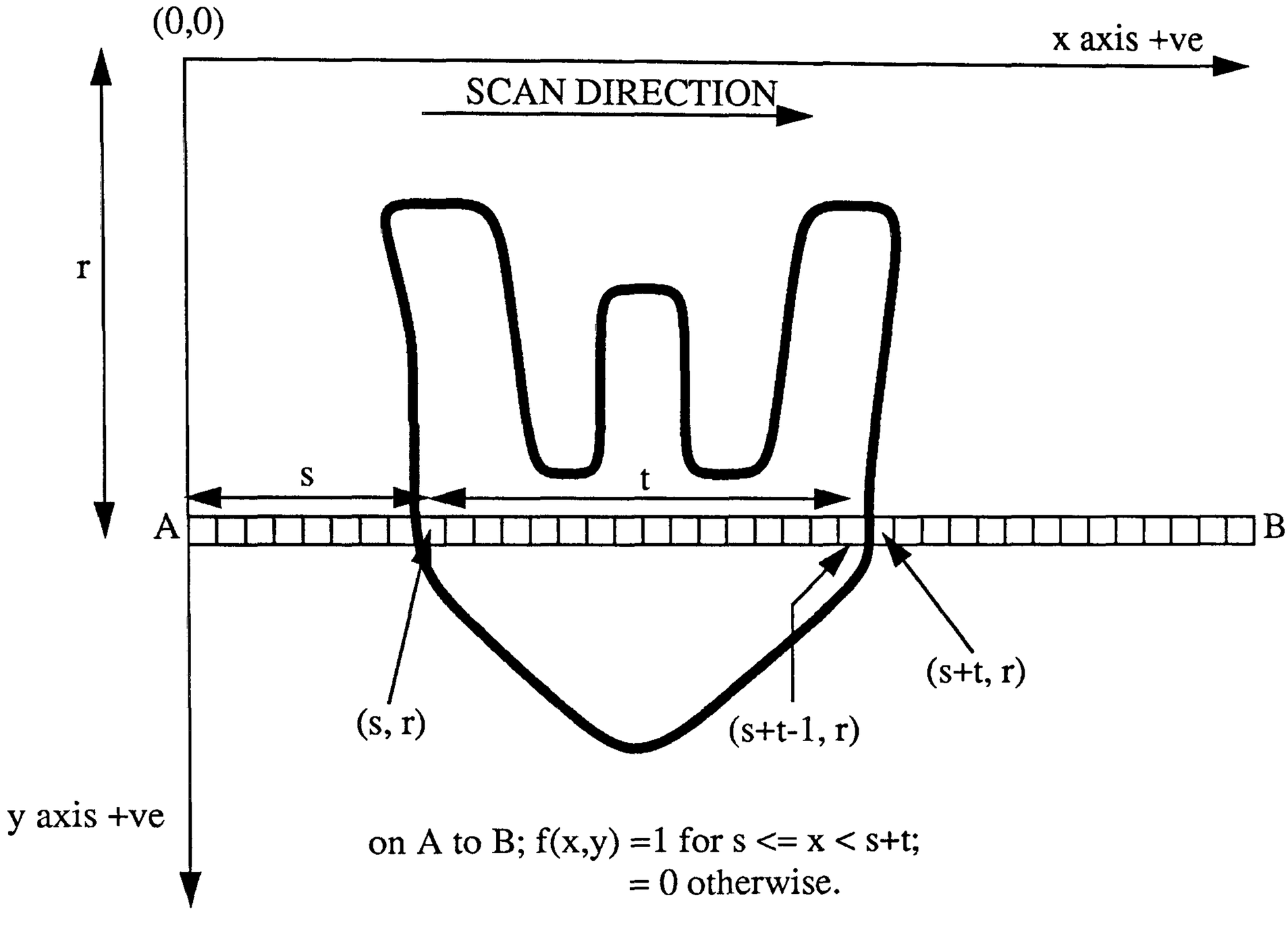


Figure 6.7 An illustration of a scanned line.

To extract the moments for each line consider the scan line A to B (in figure 6.7) which detects the section starting at pixel (s,r) and finishing at pixel (s+t, r). The length of this section within the shape is (s + t) - s = t pixels. If there are p such sections in any one scan then the area of the entire shape, μ_{00}, is given by:-

$$\mu_{00} = \sum_{\text{all } y} \sum_{\text{all } p} t \quad \text{.....eq 6.18}$$

consider the moment M_{10} (referenced to the image origin);

$$M_{10} = \sum_{\text{all } y} \sum_{all p} \sum_{x=s}^{s+t-1} x \quad \text{.....eq 6.19}$$

The finite sum of x from 1 to n is given by;

$$\sum_{x=1}^{n} x = \frac{1}{2} n(n+1) \ldots\ldots\ldots\ldots\ldots\ldots \text{eq } 6.20$$

thus equation 6.19 can be written as equation 6.21 by finding the sum to s+t-1 and subtracting the sum to s-1;

$$M_{10} = \sum_{\text{all}\,y} \sum_{\text{all}\,p} \frac{1}{2} (s+t-1)\ (s+t) - \frac{1}{2} (s-1)\, s \ldots\ldots \text{eq } 6.21$$

Using the finite sum of x^2 and x^3 given by;

$$\sum_{x=1}^{n} x^2 = \frac{1}{6} n(n+1)\ (2n+1) \ldots\ldots\ldots \text{eq } 6.22$$

and

$$\sum_{x=1}^{n} x^3 = \frac{1}{4} n^2 (n+1)^2 \ldots\ldots\ldots\ldots\ldots\ldots \text{eq } 6.23$$

the values for M_{02}, M_{03}, M_{11}, M_{12}, M_{22}, M_{21}, M_{10} and M_{30} can be calculated in a similar way;

$$M_{20} = \sum_{\text{all}\,y} \sum_{\text{all}\,p} \frac{1}{6} (s+t-1)\ (s+t)\ (2s+2t-1)$$

$$-\frac{1}{6} (s-1)\, s\, (2s-1) \ldots\ldots\ldots \text{eq } 6.24$$

$$M_{30} = \sum_{\text{all}\,y} \sum_{\text{all}\,p} \frac{1}{4} (s+t-1)^2 (s+t)^2 - \frac{1}{4} (s-1)^2 s^2 \ldots\ldots \text{eq } 6.25$$

$$M_{11} = \sum_{\text{all}\,y} \sum_{\text{all}\,p} y \left[\frac{1}{2} (s+t-1)\ (s+t) - \frac{1}{2} (s-1)\, s \right] \ldots\ldots \text{eq } 6.26$$

$$M_{21} = \sum_{\text{all}\,y}\sum_{\text{all}\,p} y\left[\frac{1}{6}(s+t-1)\ (s+t)\ (2s+2t-1)\right.$$

$$\left.-\frac{1}{6}(s-1)\,s\,(2s-1)\ \right] \ldots\ldots\ldots \text{eq } 6.27$$

$$M_{22} = \sum_{\text{all}\,y}\sum_{\text{all}\,p} y^2\left[\frac{1}{6}(s+t-1)\ (s+t)\ (2s+2t-1)\right.$$

$$\left.-\frac{1}{6}(s-1)\,s\,(2s-1)\ \right] \ldots\ldots\ldots\ldots \text{eq } 6.28$$

$$M_{12} = \sum_{\text{all}\,y}\sum_{\text{all}\,p} y^2\left[\frac{1}{2}(s+t-1)\ (s+t) - \frac{1}{2}(s-1)\,s\right] \ldots\ldots \text{eq } 6.29$$

$$M_{01} = \sum_{\text{all}\,y}\sum_{\text{all}\,p} yt \ldots\ldots\ldots\ldots\ldots \text{eq } 6.30$$

$$M_{02} = \sum_{\text{all}\,y}\sum_{\text{all}\,p} y^2 t \ldots\ldots\ldots\ldots\ldots \text{eq } 6.31$$

$$M_{03} = \sum_{\text{all}\,y}\sum_{\text{all}\,p} y^3 t \ldots\ldots\ldots\ldots\ldots \text{eq } 6.32$$

If higher order moments were required then the sum of the finite series, x^n, would have to be evaluated using a pre-calculated look up table of summation values. Expressions of the type given in equation 6.20 do not exist for n>3.

6.5.2 Moments relative to component centroid

Having calculated the moments of the image relative to the arbitrary origin they must be shifted to the image centroid thus producing translation invariant moments. Only those pixels forming part of the component's silhouette will have a value of 1, the remainder will have a value of 0. Hence the centroid of the image is also that of the component. The component centroid is given by;

$$XCOORD = \frac{M_{10}}{\mu_{00}} \quad \text{.....eq 6.33}$$

$$YCOORD = \frac{M_{01}}{\mu_{00}} \quad \text{.....eq 6.34}$$

To shift the moments from the arbitrary origin to the centroid (or any other origin) consider the following scenario in which pixel p_j is X_j away from the arbitrary y axis and x_j away from the new y axis through the component centroid. The component centroid (new y axis) is b away from the arbitrary y axis;

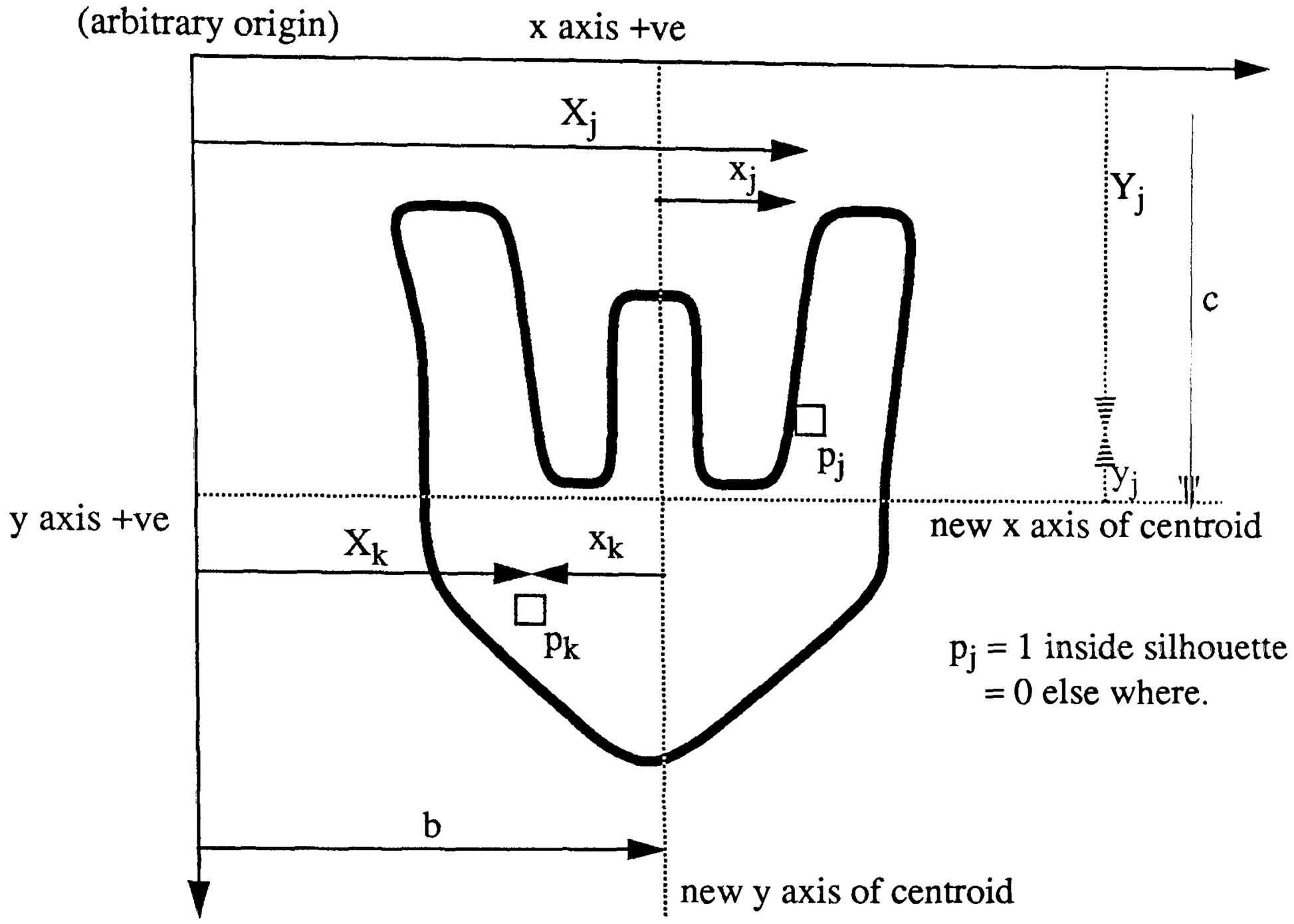


Figure 6.8 Moving the moment values relative to the shape centroid.

Moment M_{10} about the arbitrary origin must be shifted to become μ_{10} about the new centroid axis;

$$M_{10} = \sum_{\text{all}\,j} p_j X_j \ldots\ldots\ldots\ldots\ldots \text{eq } 6.35$$

$$\mu_{10} = \sum_{\text{all}\,j} p_j x_j \ldots\ldots\ldots\ldots\ldots \text{eq } 6.36$$

$$X_j = b + x_j \ldots\ldots\ldots\ldots\ldots \text{eq } 6.37$$

(Note that for p_k, x_k is negative so $X_k = b + x_k$)

$$\mu_{10} = \sum_{\text{all}\,j} p_j (X_j - b) \ldots\ldots \text{eq } 6.38$$

$$\mu_{10} = \sum_{\text{all}\,j} p_j X_j - \sum_{\text{all}\,j} p_j b \ldots \text{eq } 6.39$$

$$\mu_{10} = M_{10} - bU_{00} \ldots\ldots\ldots \text{eq } 6.40$$

μ_{20}, μ_{30}, μ_{01}, μ_{02} and μ_{03} can be similarly calculated;

$$\mu_{20} = M_{20} - 2bM_{10} + b^2\mu_{00} \ldots\ldots \text{eq } 6.41$$

$$\mu_{30} = M_{30} - 3bM_{20} + 3b^2M_{10} - b^3\mu_{00} \ldots\ldots\ldots \text{eq } 6.42$$

If the new x axis of the centroid is c below the arbitrary one then;

$$\mu_{01} = M_{01} - c\mu_{00} \ldots\ldots\ldots \text{eq } 6.43$$

$$\mu_{02} = M_{02} - 2cM_{01} + c^2\mu_{00} \ldots\ldots\ldots \text{eq } 6.44$$

$$\mu_{03} = M_{03} - 3cM_{03} + 3c^2M_{03} - c^3\mu_{00} \ldots\ldots \text{eq } 6.45$$

For the two dimensional moments consider μ_{21} and a pixel which in addition to the position described previously is also Y_j away from the arbitrary axis and y_j away from the new centroid axis. The centroid axis is c away from the arbitrary axis;

$$\mu_{21} = \sum_{\text{all}\,j} p_j x_j^2 y_j \ldots\ldots\ldots \text{eq } 6.46$$

and since

$$Y_j = c + y_j;\ X_j = b + x_j \ldots\ldots\ldots\ldots \text{eq } 6.47$$

then we have:-

$$\mu_{21} = \sum_{\text{all}\,j} p_j (X_j - b)^2 (Y_j - c) \ldots\ldots\ldots\ldots \text{eq } 6.48$$

expanding this we obtain:-

$$\mu_{21} = \sum_{\text{all}\,j} p_j X_j^2 Y_j + b^2 \sum_{\text{all}\,j} p_j Y_j - 2b \sum_{\text{all}\,j} p_j X_j Y_j - c \sum_{\text{all}\,j} p_j X_j^2$$

$$- b^2 c \sum_{\text{all}\,j} p_j + 2bc \sum_{\text{all}\,j} p_j X_j \ldots\ldots\ldots\ldots \text{eq } 6.49$$

μ_{21} can then be written in terms of the moments relative to the arbitrary origin:-

$$\mu_{21} = M_{21} + b^2 M_{01} - 2bM_{11} - cM_{20} - b^2 c\mu_{00} + 2bcM_{10} \ldots \text{eq } 6.50$$

The other two dimensional moments can be found in a similar way:-

$$\mu_{12} = M_{12} + c^2 M_{10} - 2cM_{11} - bM_{02} - c^2 b\mu_{00} + 2bcM_{01} \ldots\ldots \text{eq } 6.51$$

$$\mu_{11} = M_{11} - cM_{10} - bM_{01} + abM_{00} \ldots\ldots\ldots\ldots \text{eq } 6.52$$

The moment invariant feature set can then be calculated as detailed in section 6.3. The increase in speed is considerable. If the moments are calculated on a pixel by pixel

basis then the total time to produce a set of features for one component averages 18,000 cycles per image line. The method described here would require on average 336 cycles per image line. Unfortunately each pixel must be singularly clocked from the camera which contains 2048 pixels per line so the minimum time per line is 2048 clock cycles.

Only by increasing the computational speed in this way does the method of moments and moment invariants become viable for real time image recognition.

Note that this method for transforming the second order moments is similar to the parallel axis theorem [e.g. CHESTER 1979], which is used in mechanics for calculating moments of inertia in three dimensional objects.

6.6 Reducing the noise in the feature values

For continuous shapes that contain no noise the values of the moment features will be strictly invariant as the shape is rotated and translated. Due to the quantisation errors introduced by the image capture method and the general environmental noise this is not the case for real digital images as can be seen in figure 6.2. For the whole data base used in this work the feature values varied in practice by between 0.1% and 20% depending on the particular component. This variation can cause significant problems in terms of identifying similar shapes which have very similar feature values. Teague [TEAGUE 1980] suggests that these variations can be overcome by taking several images of each shape and averaging the moment invariant values. As a general approach this method is too time consuming for an industrial production line process such as shoe manufacture, since the time spent in obtaining extra images of a component could be used in processing other components. It is important, therefore, to minimise the variation of shape features in different images. The idea of extra images per shape has not been completely ignored and chapter 7 will investigate the use of several nominally 'identical' images as training inputs to neural networks used for recognising difficult shapes from a single image.

6.6.1 Filtering the image

If the original image is noisy then all subsequent identification processes will be handicapped. Images can be filtered to remove noise and many algorithms are available for this purpose [e.g. NIBLACK 1986]. The algorithm used will be chosen so that the image is altered to suit the requirements of the subsequent processes. The general approach is that the centre pixel of a square grid of pixels is modified as according to a

function of its neighbours. In the case of leather shoe upper components the environmental noise problem includes the following identifiable causes:-

- Loose leather fibres.

- Thin fibres of leather attached to the cut edge of components.

Tout and BUSM devised methods of reducing these effects as detailed in chapter 3 and summarised here:

Firstly by scanning the component over an air gap any loose fibres fall away and are removed from the image. Secondly thresholding each pixel each pixel removes those pixels which are only partially covered and likely to be noise. Finally the scanned lines are passed through a three pixels by three pixels matrix which extracts all the edge point pixels as defined in 2.2.5. Most loose fibres are less then a pixel in width (although they may be several pixels long) and are removed from the image along with the shape's internal pixels leaving just the edge points.

As part of this current research the choice of matrix size was examined to see if an improvement in filtering performance could be achieved. Note that if moment invariants are to be used as features then it is no longer necessary to extract the edge points and any pixel which is part of a corner group can 'pass' through the filter. The filter equation (e.q. 2.2) can be simplified to:

$$P(i,j) = A + B + C + D \ldots\ldots\ldots\ldots \text{eq } 6.53$$

The following sizes of the filter matrix centred on P(i,j) were tried:- 1 pixel by 1 pixel, 3 pixels by 3 pixels, 5 pixels by 5 pixels, 7 pixels by 7 pixels and 9 pixels by 9 pixels, with each being applied to several examples of images. Filters with even numbers of coefficients would give different results according to the rotation of the component and were not tried. A similar argument would apply to filters having differing numbers of rows and columns. The variance for each moment invariant feature over the tests was obtained and the results can be seen in figure 6.9.

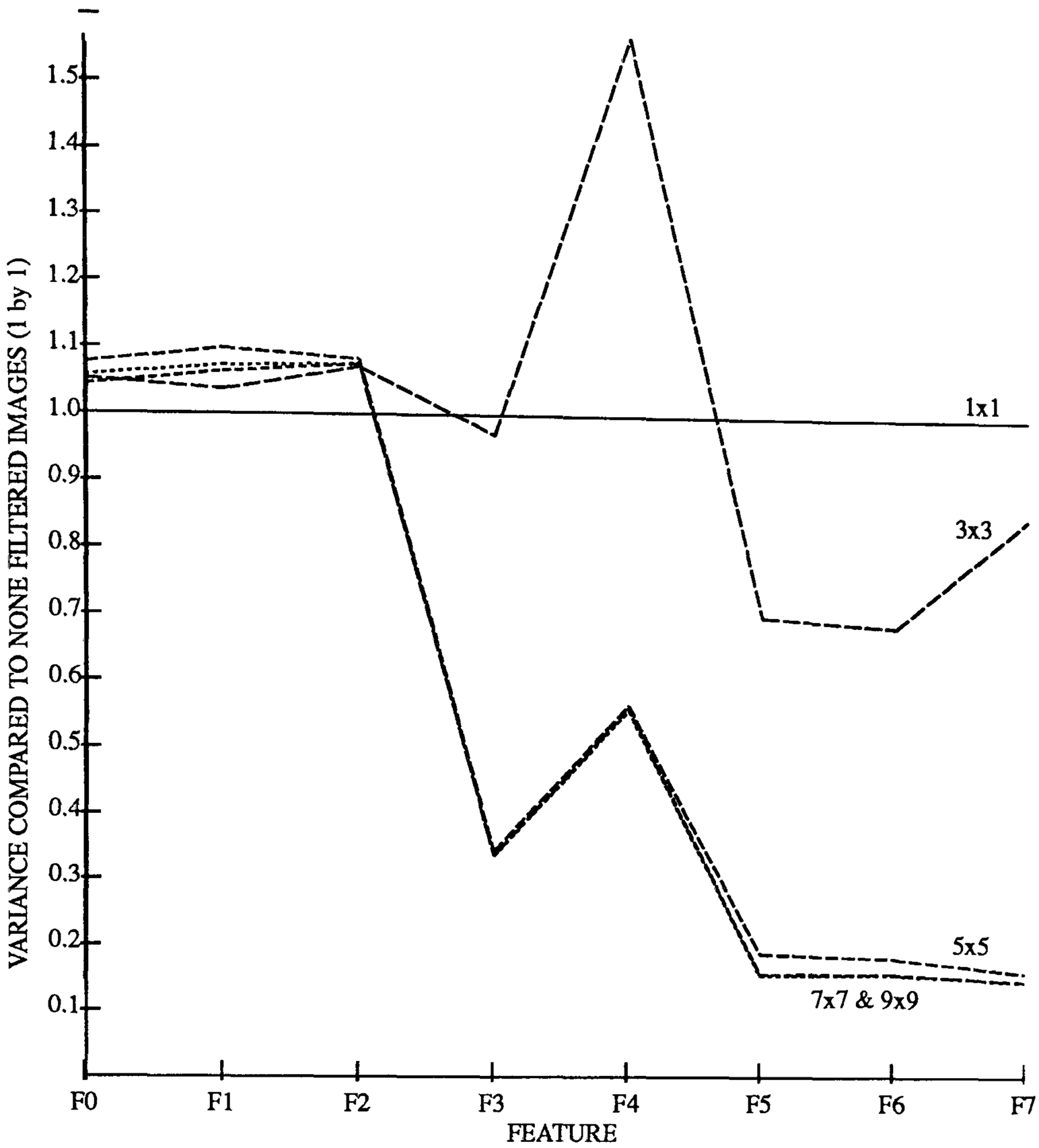


Figure 6.9 Performance variation with filter matrix size.

The use of a 5 pixels by 5 pixels matrix was judged to give the optimum performance for the grid sizes tried. It produced a lower variance than the three by three filter and performed equally as well as the seven by seven and nine by nine filters. The five by five filter has the obvious advantage of requiring less computation time or hardware.

The DSP/Transputer system designed by BUSM was intended to provide all the signal processing functions for the vision module. Despite the generation of filtering algorithms to work on the compressed data the required speed could not be achieved. Thus filtering (if used) will have to be performed in hardware, unless the processing power is increased.

6.6.1.1 Recognition performance with 5 by 5 filter

It is impractical to perform frequent tests with a 'full size' data base approaching 20,000 components. Most systems can quite easily separate components into family groups. The test sets used in these experiments were based on such closely related

shapes, containing just 20 component types consisting of five groups and three single components as detailed in appendix IV.

The average performance of the moment invariant feature system utilising a 5 by 5 pixel filter to pre-process the image lines is shown in table 6.3. The pattern recognition method used is detailed in section 6.4.1.3.

TRUE SHAPE \ ALLOCATED SHAPE	Tsh01	Tsh02	Tsh03	Tsh05	Tsh06	Tsh08	Tsh09	Tsh10	Tsh11	Tsh12	Tsh13	Tsh14	Tsh15	Tsh16	Tsh19	Tsh20	Tsh21	Tsh23	Tsh28	Tsh38	REJECT
Tsh01	90.6																				9.4
Tsh02		100																			
Tsh03			90.6																9.4		
Tsh05				100																	
Tsh06					96.1																3.9
Tsh08						97.2															2.8
Tsh09							74.4							25.6							
Tsh10								100													
Tsh11									91.1	8.9											
Tsh12									7.2	91.7			1.1								
Tsh13									1.7		98.3										
Tsh14												97.8									2.2
Tsh15									2.8	0.6			96.6								
Tsh16							25.2	2.2						70.0	2.2						0.4
Tsh19															100						
Tsh20																97.2					2.8
Tsh21																16.1	61.2	13.3			9.4
Tsh23																1.1	2.2	96.7			
Tsh28			4.4																90.0		5.6
Tsh38																				100	

Table 6.3 Average recognition results for images pre-processed using a 5 by 5 filter

To summarise;-

The average substitution rate (components allocated to wrong class) is 6.20%

The average rejection rate (components unidentified) is1.83%

Average total failure rate (sum of substitution and rejection errors) is 8.03%

The summary results are obtained by adding all the (e.g. substitutions) together, dividing by 2000 (the total number of tests) and multiplying by 100%.

Table 6.3 shows the results for a group of components selected to give difficulties in recognition and as such gives a very pessimistic view of the algorithm performance. In normal operating conditions the data base would contain many more components which are easily identifiable. The overall differences between performance using filtered images (table 6.3) and non-filtered images (table 6.2) are minimal due to the lack of dust type noise present in the original images. It is possible that the low pass filtering operation (averaging) removes some detail from the component edge which might otherwise be useful for identification. This would account for the increased level of rejection in table 6.3 and the increased substitution for some individual components.

6.6.2 Compensating for the grid errors

Obtaining a digital image of a component imposes a rectangular grid onto the shape and samples the light level in each rectangle. Any fine detail which is of a smaller dimension than the grid rectangles will be lost as it is averaged with its neighbourhood. In computer systems the value for each rectangle is further digitised leading to quantisation errors in the image.

The use of rectangular sampling zones has an added complication when used to measure rotating items since a line of a given length will seem to have differing values depending upon its alignment with the grid. In figure 6.10 line a, b and c have apparent lengths of 12, 4.9 and 4.5 pixels despite being of identical real length.

The scanning system used in this research was set up with an aspect ratio of 1. That is the x axis and the y axis dimensions of each pixel are identical. If this is the case then lines (a) and (b) in figure 6.10 will be of the same apparent length. As the line is rotated its shortest value will be at an angle of 45 degrees to the axis when it will have an apparent length of $^1/\sqrt{2}$ of its axial length.

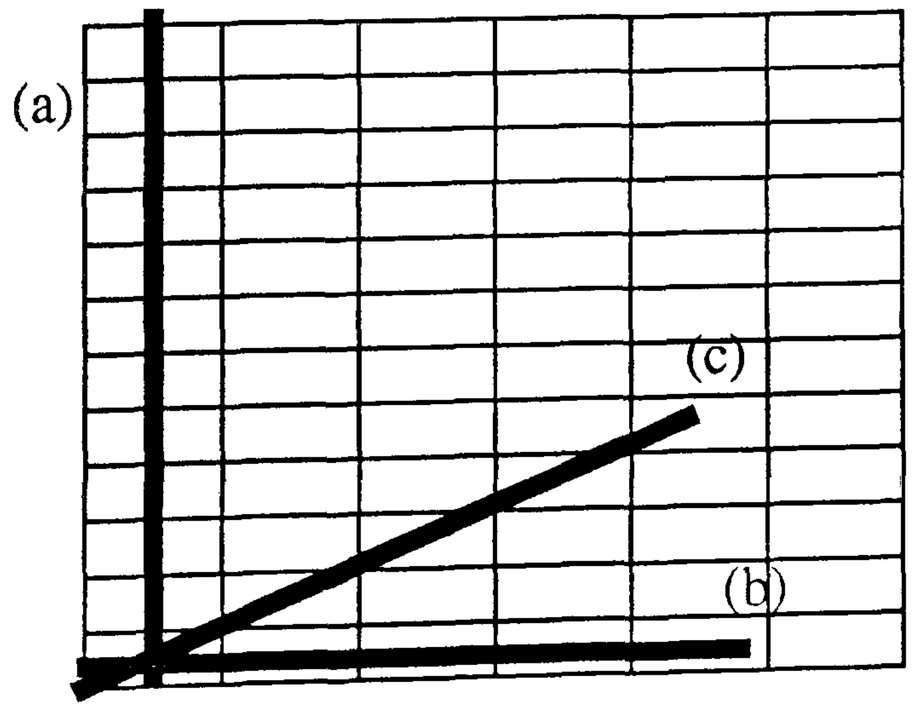


Figure 6.10 Apparent change in line length.

The variation in line length with rotation affects the values of the feature measurements and is a major contributor to feature variation. The cross points in figure 6.11 show the error (defined in equation 6.54) of Feature F2 for component Tsh14 as it is rotated through 360 degrees. The value of F2 varies with the orientation (θ) and the error for a given orientation can thus be predicted. The curve in Figure 6.11 represents the third order polynomial equation given in equation 6.55 which was used to model the error for Feature F2 of component Tsh14.

$$\text{Error} = \left(\frac{F2(\theta) - \overline{F2}}{\overline{F2}}\right) \times 100\% \ldots\ldots\ldots\ldots \text{eq } 6.54$$

$$\text{Tsh14 F2 Error Model} = a + b\theta^1 + c\theta^2 + d\theta^3 \ldots\ldots\ldots \text{eq } 6.55$$

Where

a=0.416890e+00

b=6.590278e-02

c=-1.554908e-04

d=-8.493680e-06

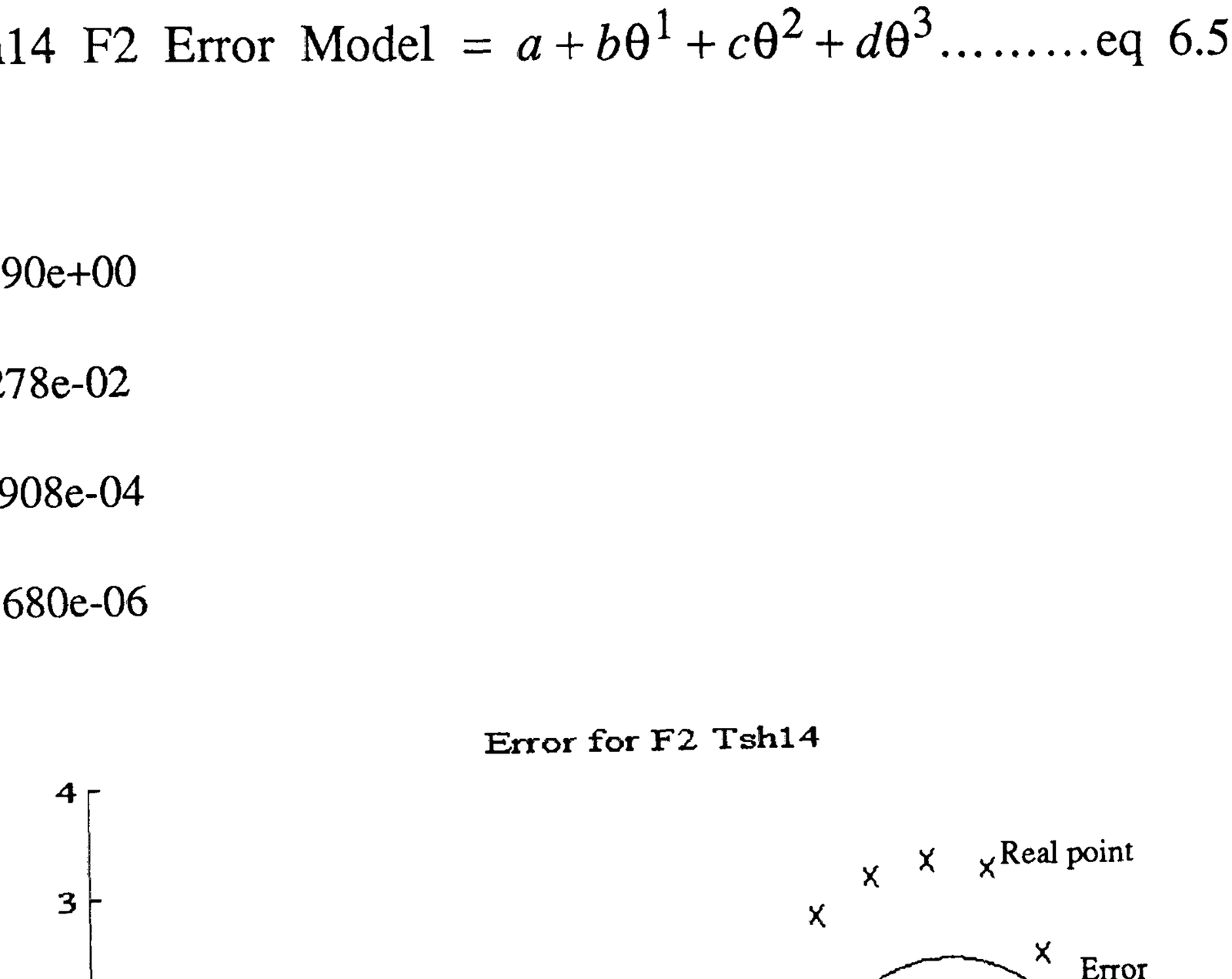


Figure 6.11 Error versus orientation (in degrees).

Using this model to correct the error decreases the spread of the feature values as shown in figure 6.12. Equation 3.6 is first used to ascertain the components orientation and equation 6.55 to determine the error. The feature value is then adjusted accordingly.

The variance for Tsh14 F2 is improved by about 87%.

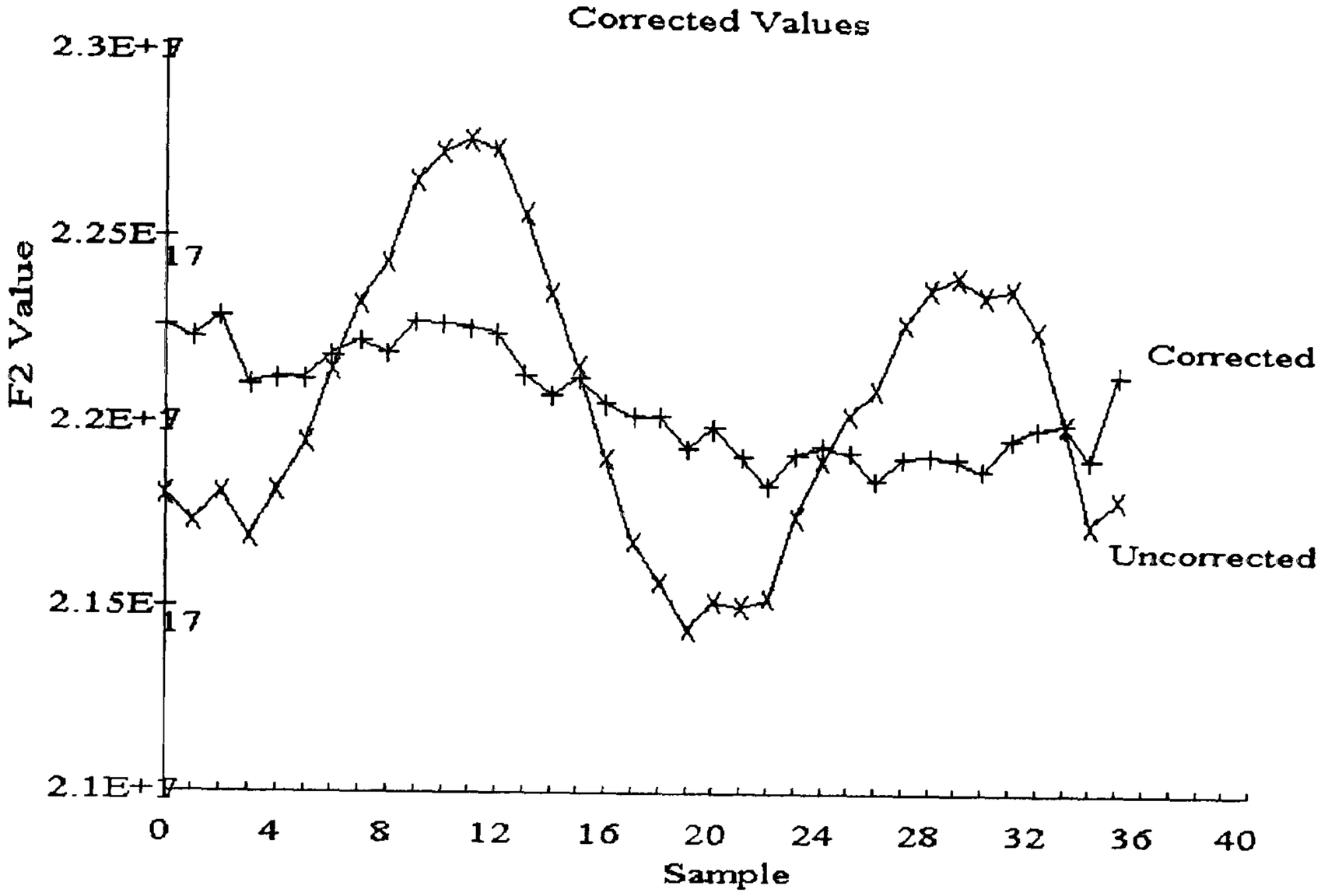


Figure 6.12 Corrected and uncorrected values for Tsh14 F2.

When the same analysis is applied to other components it is found that the shape of the error curve and its position relative to θ are the same, only the amplitude of the curves varies. It was expected that the maximum amplitude of the error would vary depending upon the relative circularity of the shapes. The GCOEFF value of a component indicates the roundness of its equivalent ellipse and was used as a measure of each components circularity. Figure 6.13 shows the peak amplitude of the F2 error curves against the GCOEFF value for several component shapes. The results indicate that as the GCOEFF value for a shape approaches zero (a circle) the variation in feature F2 becomes very large. This relationship can be modelled by equation 6.56, which is also shown on figure 6.13 and was obtained by fitting a geometric regression curve, with parameters λ and γ, to the known data. This result can be compared to the threshold used by BUSM for determining the suitability of equation 3.6 for orientation assessment.

Feature F2 and equation 3.6 have common terms in μ_{20}, μ_{02} and μ_{11}. The threshold derived experimentally by BUSM is a GCOEFF value of 0.2 and as can be seen from figure 6.13 this corresponds to an error of 4%.

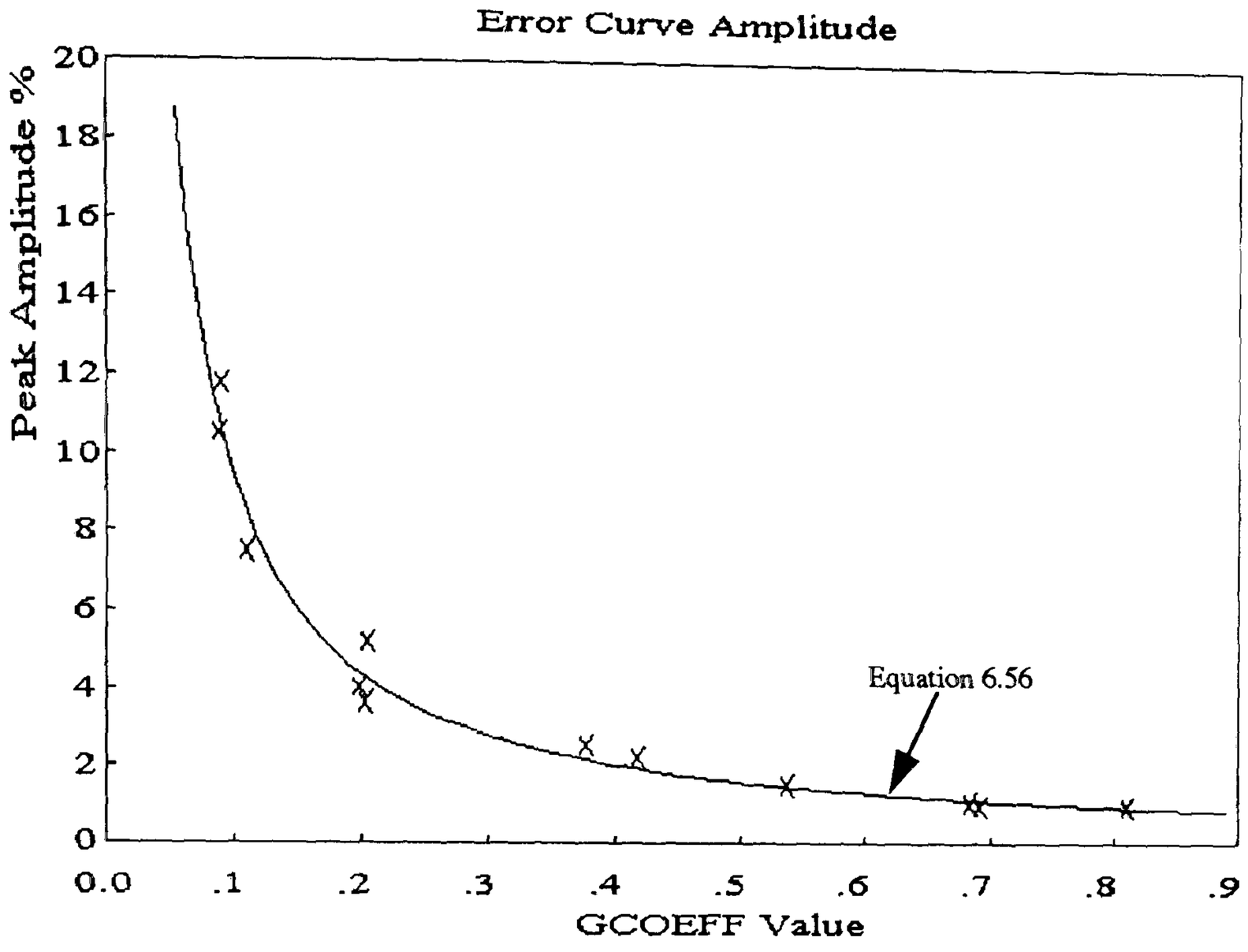


Figure 6.13 Peak amplitudes of error curves versus GCOEFF value.

$$\text{Amplitude Model} = \lambda + (GCOEFF)^{\gamma} \ldots\ldots\ldots\ldots \text{eq } 6.56$$

Where;

λ=0.7236342

γ=-1.115106

This general relationship between error, theta and GCOEFF holds true for all the features used. By using an error curve with a maximum amplitude of ±1.0 to calculate the error relative to θ and then equation 6.56 to modify the amplitude relative to GCOEFF the error can be corrected on a general basis. The third order polynomial error model for F2 with an amplitude of 1.0 is shown in equation 6.57.

$$\text{Unity Error Model} = a + b\theta^1 + c\theta^2 + d\theta^3 \ldots\ldots\ldots\ldots \text{eq } 6.57$$

where;

a=0.3558381

b=2.939372e-02

c=-1.376435e-04

d=-4.350381e-06

Thus the overall correction for a component with a given GCOEFF value at an orientation θ is given by;

$$\text{Cor} = \left[a + b\theta^1 + c\theta^2 + d\theta^3\right]\left[\lambda + (GCOEFF)^{\gamma}\right] \ldots \text{eq } 6.58$$

with a, b, c, d, λ and γ are as defined previously.

The values of a, b, c, d, λ and γ for the other features are shown in appendix V.

6.6.2.1 Recognition performance with grid error correction

The average performance of the moment invariant feature system utilising a 5 by 5 pixel filter and grid error correction is shown in table 6.4. The pattern recognition method used is detailed in section 6.4.1.3.

TRUE SHAPE \ ALLOCATED SHAPE	Tsh01	Tsh02	Tsh03	Tsh05	Tsh06	Tsh08	Tsh09	Tsh10	Tsh11	Tsh12	Tsh13	Tsh14	Tsh15	Tsh16	Tsh19	Tsh20	Tsh21	Tsh23	Tsh28	Tsh38	REJECT
Tsh01	100																				
Tsh02		100																			
Tsh03			87.8																12.2		
Tsh05				100																	
Tsh06					96.1																3.9
Tsh08						97.2															2.8
Tsh09							90.0							10.0							
Tsh10								100													
Tsh11									92.2	6.7			1.1								
Tsh12									2.2	97.8											
Tsh13									2.5		97.5										
Tsh14												100									
Tsh15									0.6				99.4								
Tsh16							16.1	2.2						78.7	3.0						
Tsh19															100						
Tsh20																97.2					2.8
Tsh21																15.6	58.8	15.0			10.6
Tsh23																1.1	3.3	95.6			
Tsh28			5.6																88.8		5.6
Tsh38																				100	

Table 6.4 Average recognition results for features adjusted using equation 6.58.

To summarise;-

Substitution rate=4.86%

Rejection rate=1.29%

Overall failure rate=6.15%

As with table 6.3 these results summarise the performance with a set of components chosen as a difficult identification challenge and as such presents a pessimistic result of the likely performance on a 20,000 component database. Compensating the features for the error caused by the orientation of the imaging grid reduced the substitution rate, the rejection rate and thus the overall failure rate from 8.03% to 6.15%.

6.7 Summary

It was already known due to the earlier work of Tout [TOUT 1989] that low order moments were suitable for locating shoe components and that in certain circumstances

they could also be used to determine component orientation. This chapter has shown that moments can also be used to identify shoe components of considerable similarity and that higher order moments can be used to determine component orientation. A complete identification system can thus be formed utilising only moment based features.

Tests on similar shapes of equal area confirmed the use of moment invariants for discriminating between shapes and showed that moments could be used as the basic feature set for describing the shoe components. These tests did, however, highlight the noise problems associated with moments and considerable effort was put into reducing these noise effects.

The results for the low pass filter were discouraging since the averaging effect seemed to remove essential details required to distinguish very similar components. The 5 by 5 pixel matrix decreased feature variation for single shapes but has an increased smoothing effect compared with smaller filter areas. The results for the grid compensation were particularly encouraging since a significant improvement (23.13%) in overall failure rate was achieved with no additional shape sampling. The neural networks reviewed in chapter 7 give the best increase in performance but require additional samples of prototype shapes which is a time consuming process.

In addition to the discriminating abilities of moments, thought had to be given to the processing power required. Calculating moment invariants on a pixel by pixel basis is an intense task and only by devising methods which allow the moment calculations to be performed on line with compressed data has the method become practical.

CHAPTER 7

CLASSIFICATION BY NEURAL NETWORKS

7.1 Introduction

Recognition errors are caused when the features of the working component differ significantly from those of the appropriate prototype. If the features lie within the realms of another prototype then substitution will occur otherwise the component will be rejected. The value of the prototype features relevant to the average value of the class will considerably influence the number of recognition errors.

Tout allowed for the noise variation of his features by the use of tolerances when comparing components and prototypes. While this method worked equally as well for moment features it proved hard to minimise the number of errors by optimising the tolerance values. Teague [TEAGUE 1980] overcame the problem of noise variation for moment invariants by using 20 images of both the prototype and the working shapes in order to average the feature values. The acquisition of more than one image of the working components is not practical in a simple production line system due to the effects on rates of production. It is feasible (but undesirable) to use several images of prototype components to produce a more representative feature set.

An alternative to averaging the features of several example shapes was to use them as inputs to a neural network [e.g. HECHT-NIELSEN 1991] and achieve improved recognition by nonlinear separation of the feature set. Two methods were tried. The first used several examples of each class and its nearest neighbours to train a network to recognise particular classes with a view to separating the more difficult class groups such as Tsh09 and Tsh16. This method would require an operator to train the networks as and when 'difficult' shape groups are encountered. The second method trained a network to recognise pairs of matching feature sets and would need no additional operator input once installed, effectively ascertaining the feature match tolerance values automatically.

7.2 Neural networks

Neural networks are used to partition sets of data in order to distinguish them. They are particularly useful in pattern recognition tasks where their ability to generalise allows them to cope with factors such as noise or image distortions. In this research the components to be identified have been represented by transitional and orientation invariant features but due to the quantisation effects of the image acquisition a considerable amount of 'noise' still exists in the features. The use of Neural Networks may be able to overcome this.

7.2.1 General uses of neural networks

The human brain consists of several billion interconnected neurons which are able to make nonlinear decisions. The artificial neural network interconnects several neurons to achieve similar nonlinear behaviour. The resulting systems perform very well in tasks of pattern association when the patterns to be recognised are not definite. Neural networks have been used for a wide variety of tasks, from the prediction of financial markets [KANE 1994 *et al.*] to the inspection of knitted lace [SANBY 1995 *et al.*]. Petrosino and Salvi [PETROSINO 1994 *et al.*] review some applications of neural network theory to the recognition of two dimensional shapes. Zigmann and Saulnier [ZIGMANN 1993 *et al.*] have applied neural network recognition to the identification of two dimensional aluminium profiles.

7.2.2 The human neuron

Neural networks are based on models for the simulation of the human brain. They can be constructed in either hardware or software. In both cases they consist of a number of interconnected artificial neurons which model the activity of the biological neuron. The human brain exhibits several characteristics such as ability to learn, generalisation and association which are useful for pattern recognition. It is these attributes which are modelled in pattern recognition neural networks. The actual workings of the brain are not completely understood but for simulation purposes the following will suffice:

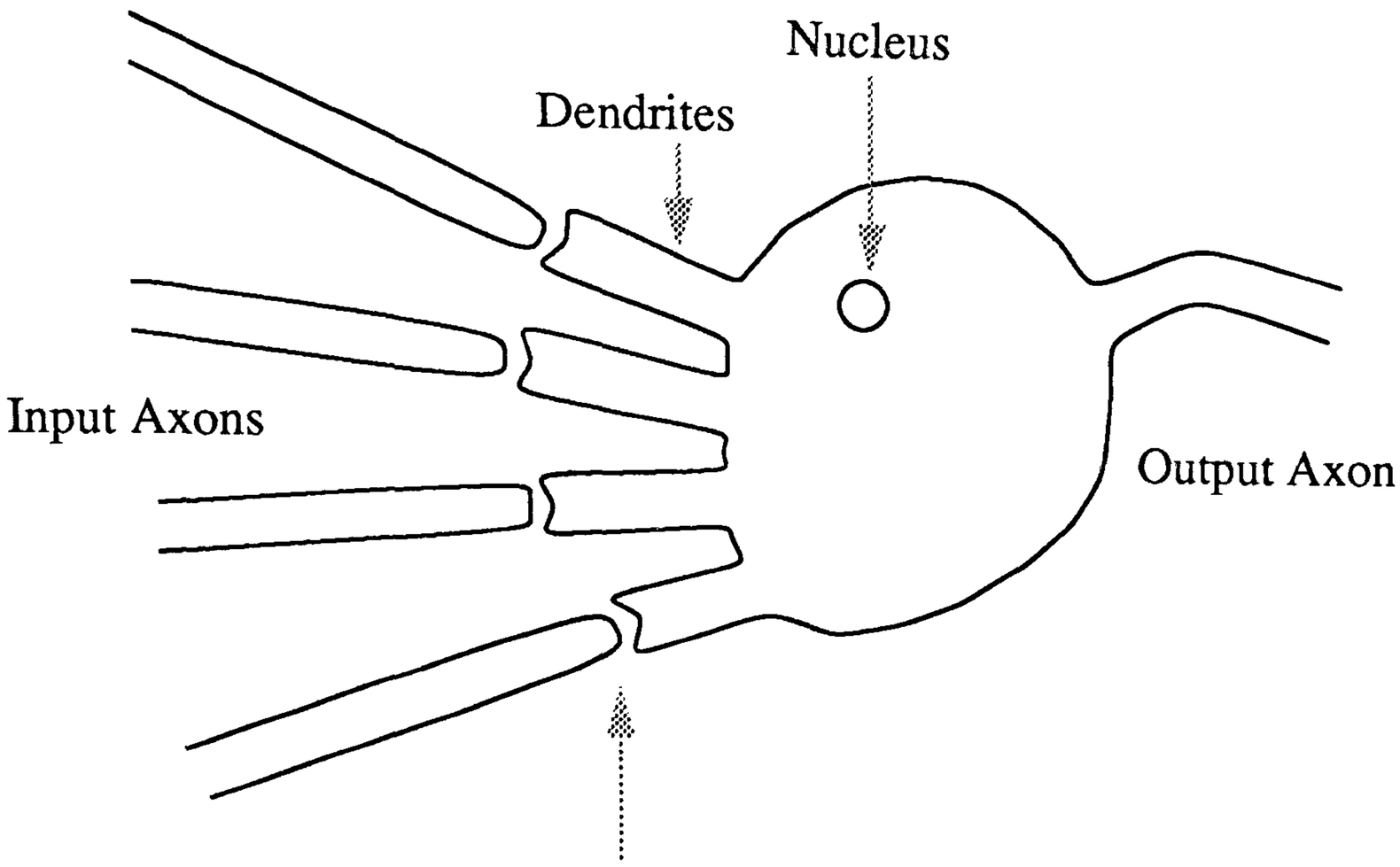


Figure 7.1 A biological neuron.

Biological neurons consist of axons (inputs cells) which feed signals into the nucleus of the neuron via the synapse and the dendrites. Each synapse attenuates the corresponding input signal according to its perceived importance as evolved through experience and learning. The signals received by the nucleus through the dendrites are then combined to give an over all representation of the input signals. If the strength of the combined signal is sufficient then the neuron will 'fire', sending a signal along its output axon to neighbouring neurons. In this way signals are propagated through the brain and a decision is reached. The level of combined signal required to 'fire' the neuron is also determined through experience and learning.

In addition to the electrical propagation described above, it is thought that the brain also employs a chemical mechanism whereby active neurons release a chemical signal which spreads radially stimulating adjacent, unconnected neurons.

7.2.3 The artificial neuron

There are many different arrangements which have been developed for simulating the workings of the brain using artificial neural networks. They can be roughly divided into three groups: Supervised training, graded training and unsupervised training. In the case of the supervised algorithms the network is trained to associate a given input with a given output response. For graded training the network is given an objective and then given a measure of its success as training proceeds. For unsupervised training the networks are only provided with inputs and are left to make their own output

classification. The networks used in this investigation consist of layers of interconnected perceptrons linked together and trained under supervision.

The basic artificial neuron as first described by McCulloch and Pitts in 1948 is called a perceptron and models the biological neurons as mathematical operations. The attenuation of the synapse is replaced by a simple weighting operation performed on the inputs. The combination of signals and the 'firing' of the nucleus are modelled as a summation and threshold operation. The weights and thresholds are determined during a series of operations known as the teaching or training phase when inputs and required outputs are used to determine the values of the weights.

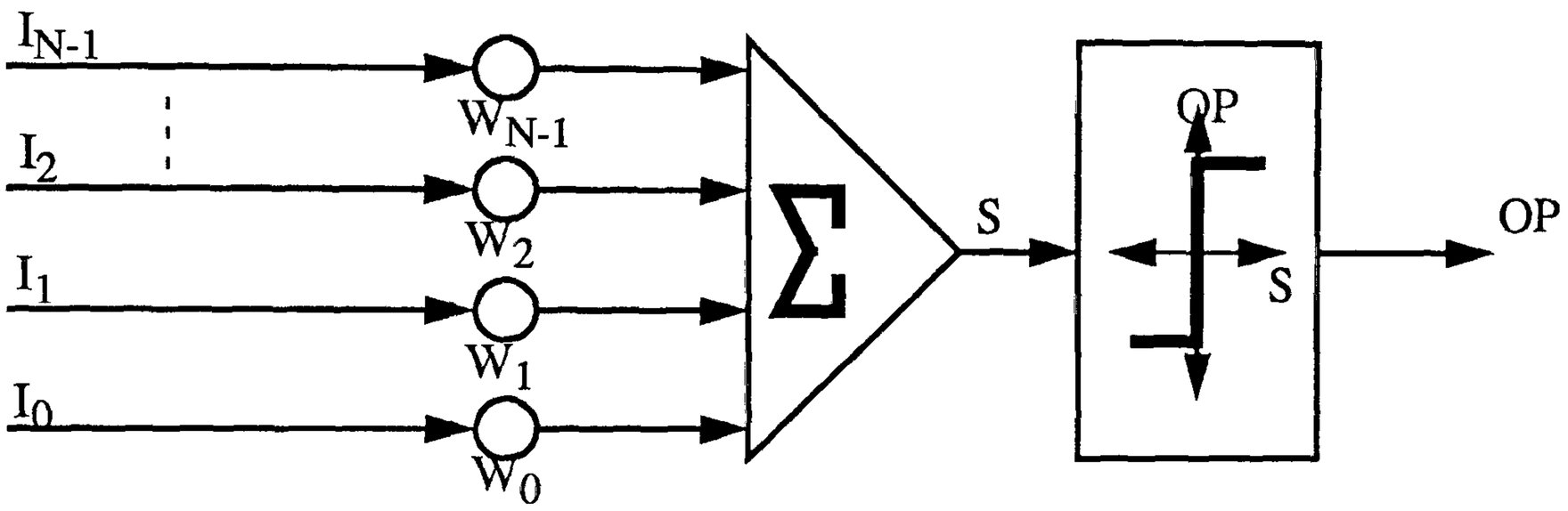


Figure 7.2 A McCulloch and Pitts neuron

The result from the summation calculation is thresholded to give a value between -1 & +1. The choice of threshold has considerable influence on the performance of the overall network.

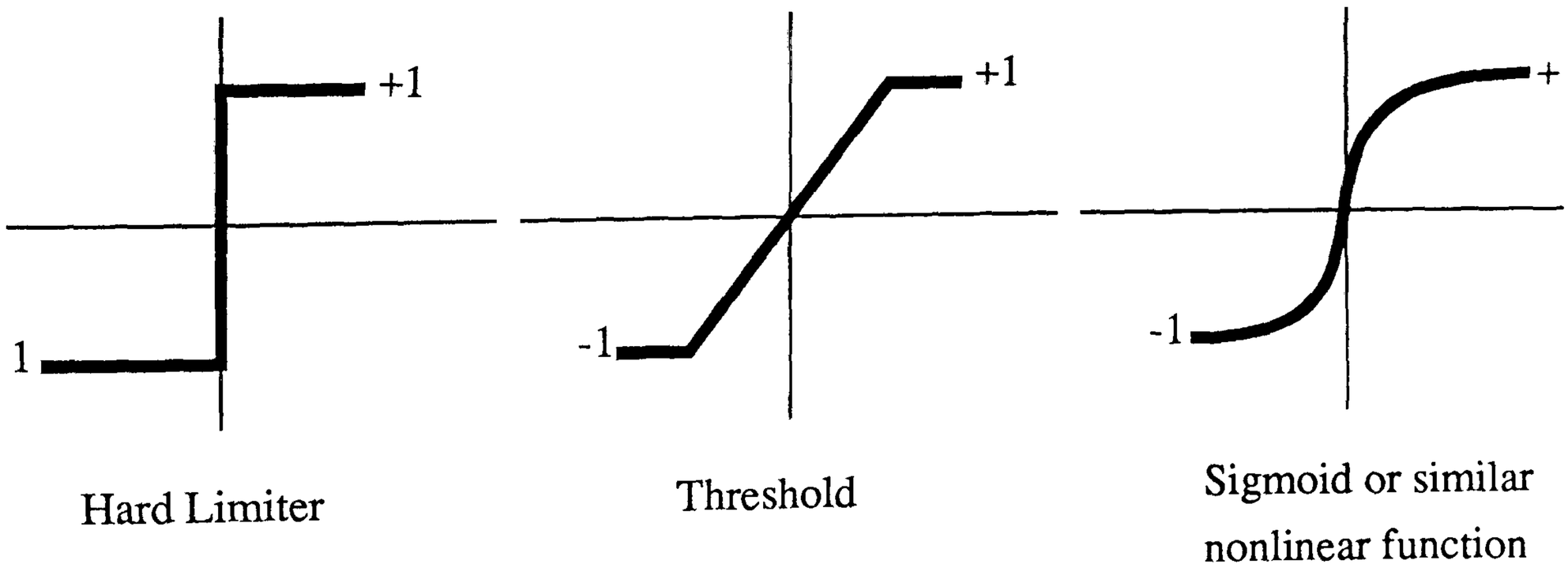


Figure 7.3 Possible types of threshold.

It is normal for the neuron to operate in a nonlinear fashion so the sigmoid function (equation 7.2) is preferred to the standard linear thresholds. The sigmoid is also easily differentiable which is of considerable advantage in the application of the Back Propagation algorithm reviewed in section 7.2.

Each neuron can be described;

$$S = \sum_{n=0}^{N-1} w_n i_n \ldots\ldots\ldots\ldots \text{eq } 7.1$$

This sum S is then passed through the threshold function (sigmoid) associated with that neuron to give its output OP.

$$OP = \frac{1}{1 + e^{-kS}} \ldots\ldots\ldots\ldots \text{eq } 7.2$$

The constant, k, controls the spread of the function.

7.2.4 Training a single McCulloch and Pitts neuron

To train a single perceptron requires several known input and output pattern pairs so that the weights can be adjusted to give the required performance. The rule used for training is:-

For a given input vector $\bar{i}$ calculate the output of the perceptron for a random set of weights ($\bar{w}$). If the output pattern is not correct modify the weights ($\bar{w}$')according to:

$$\bar{w'} = \bar{w} + C\bar{i} \ldots\ldots\ldots\ldots \text{eq } 7.3$$

where

$$C = \frac{-2S}{\bar{i} \bullet \bar{i}} \ldots\ldots\ldots\ldots \text{eq } 7.4$$

The calculation of the output of the perceptron and the weight modification continues until the required performance, in terms of output for given input, is achieved.

7.2.5 Combining neurons to form a network

Although a single perceptron can be used to solve the simplest of problems it is usual to find them linked together to form a network or neural network. These networks are of the general form shown in figure 7.4. The values obtained from each level of perceptrons are then propagated through the network. The final or output layer of a network may consist of one or more perceptrons depending upon the type of task that has been set.

Kolmogorov's representation theorem shows that multi level neural networks can be reduced to a minimal form consisting of three layers, identified as the input layer, the hidden layer and the output layer. (A discussion of Kolmogorov's theorem can be found in HECHT-NIELSEN 1991).

The training of such a network is more complex than the training of single neurons and many methods have been devised. The basic network training algorithm for supervised learning is the Back Propagation method.

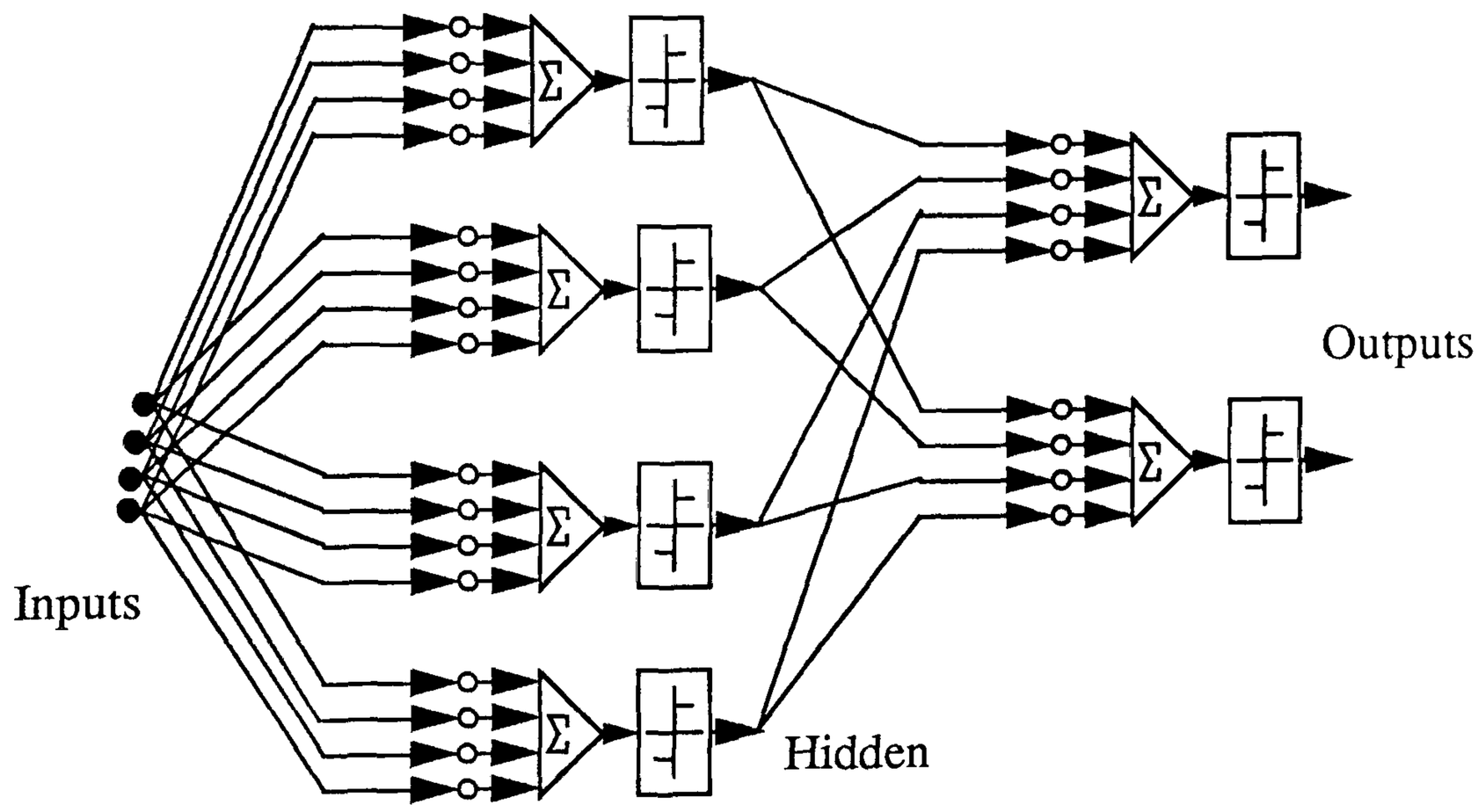


Figure 7.4 A Network of neurons.

7.3 The Back Propagation method

The process of training a network requires the adjustment of the node weights in order to get a known output for a given set of inputs (back propagation). In the case of a single perceptron both the inputs and the outputs can be compared with their required state. For multi layer networks the outputs of the output layer nodes can be measured

directly and compared with the required values. However, the values of the middle layer nodes require a more complex approach. Although intermediate node outputs are measurable the values which they should be taking are not easily established, thus they are referred to as hidden nodes.

The aim during the training phase is to reduce the error between the actual output and the required output for each training pattern (p) given by equation 7.5:-

$$E_p = 0.5 \sum_{allj} (t_{pj} - o_{pj})^2 \ldots\ldots\ldots\ldots \text{eq } 7.5$$

Where

t_{pj} and o_{pj} are the target output and actual output of node j (output nodes only) for input pattern p.

For training purposes the average error (E) over all the training patterns is obtained:

$$E = \frac{\sum_{\text{all patterns}} (E_p)^2}{\text{Number Training Patterns}} \ldots\ldots\ldots\ldots \text{eq } 7.6$$

In order to update the weights of such a network the following method is used; [e.g. RITTER (1992) *et al.*]

Consider the following figure which represents the j^{th} node or perceptron in the network. The inputs to this node are the outputs of nodes in previous layers and the output of this node will be an input to a node in a subsequent layer. The additional bias input may be added to increase the dimensionality of the network and improve the separation of the input patterns.

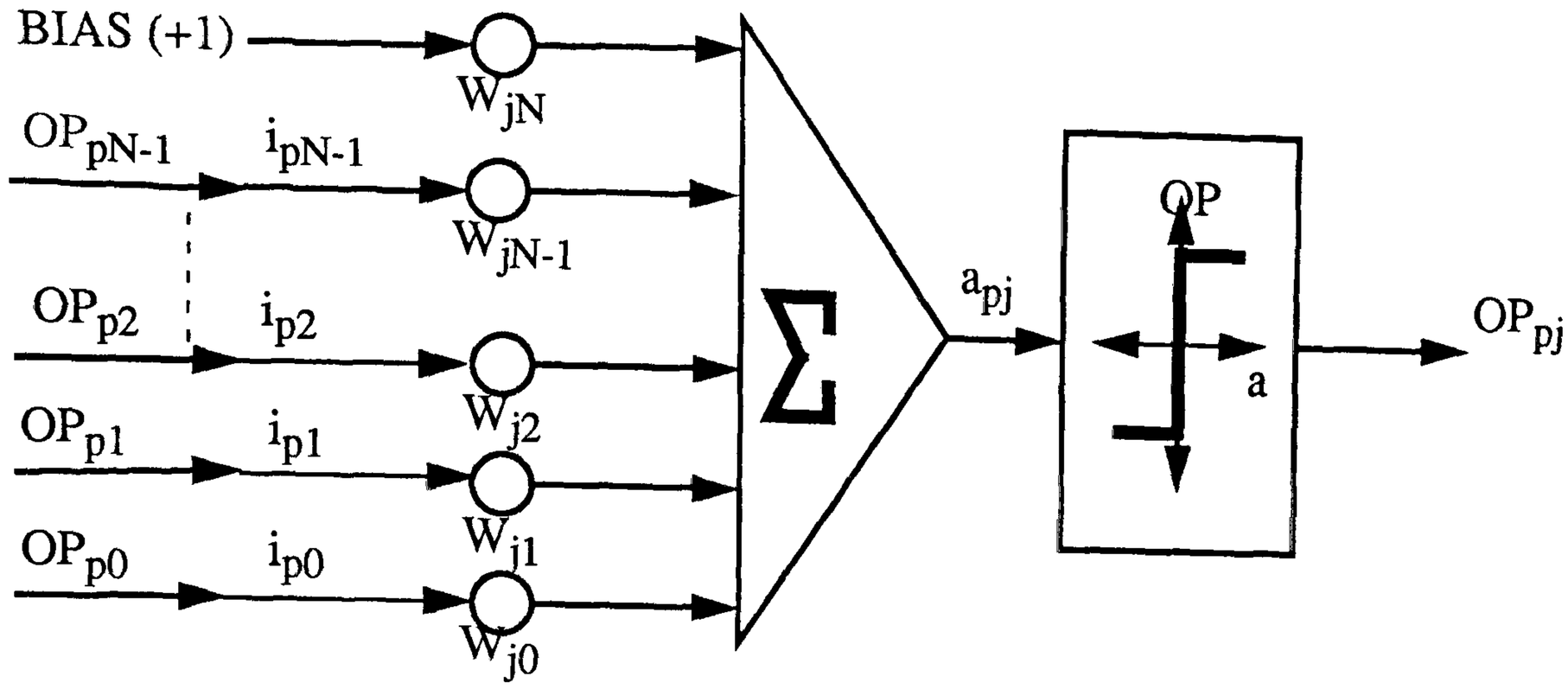


Figure 7.5 A Neuron with interconnections.

The sum of the inputs is known as the activation value (a_{pj}) applied to the threshold (f_j)

$$a_{pj} = \sum_{\text{all } i} w_{ji} o_{pi} + w_{jN} \ldots\ldots\ldots\ldots \text{eq } 7.7$$

The output of the j^{th} node for pattern p is;

$$o_{pj} = f_j(a_{pj}) \ldots\ldots\ldots\ldots\ldots\ldots \text{eq } 7.8$$

In order to train the network the weight vector for the j^{th} node must be altered thus;

$$w_{ji\,(t+1)} = w_{ji\,(t)} + \Delta_p w_{ji\,(t)} \ldots\ldots\ldots\ldots \text{eq } 7.9$$

Where;

$$\Delta_p w_{ji\,(t)} = \beta \partial_{pj} o_{pj} \ldots\ldots\ldots\ldots\ldots\ldots \text{eq } 7.10$$

β is the learning rate and controls the size of the incremental adjustments to the weights.

If the j^{th} node is the output node then the error ∂_{pj} for pattern p is defined as;

$$\partial_{pj} = (t_{pj} - o_{pj}) f'(a_{pj}) \ldots\ldots\ldots\ldots \text{eq } 7.11$$

If the j^{th} node is a hidden node then the error for pattern p is defined as;

$$\partial_{pj} = \left[\sum_{\text{all}\,k} \partial_{pk} w_{kj\,(t)}\right] f'_j(a_{pj}) \ldots\ldots\ldots\ldots \text{eq } 7.12$$

The activation function used in this case is the sigmoid function;

$$op_j = \frac{1}{1+e^{-ka_{pj}}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{eq } 7.13$$

Thus

$$f'(a_{pj}) = \frac{do_{pj}}{da_{pj}} = o_{pj}(1-o_{pj}) \ldots\ldots \text{eq } 7.14$$

The error for an output node is;

$$\partial_{pj} = (t_{pj}-o_{pj})\,(o_{pj}(1-o_{pj})) \ldots\ldots \text{eq } 7.15$$

The error for a hidden node is;

$$\partial_{pj} = \left[\sum_{\text{all}\,k} \partial_{pk} w_{kj\,(t)}\right] (o_{pj}(1-o_{pj})) \ldots\ldots\ldots\ldots \text{eq } 7.16$$

where

$$\sum_{\text{all}\,k} \partial_{pk} w_{kj\,(t)}$$

is the sum of the errors of the next layer nodes (output nodes in a three layer network) multiplied by the corresponding connecting weights prior to any adjustment.

The network is trained using a set of known input and output vectors which are used to obtain the average network error as given in equation 7.6. The weights for each neuron

can then be adjusted by working backwards from the output layer using equations 7.9, 7.10, 7.15 and 7.16. The training process is iterative and the weights are adjusted until a required error level is achieved.

7.3.1 Momentum rate

The Back Propagation algorithm for establishing the weight vector has two disadvantages. Firstly it adjusts the weight values in order to minimise the output error of the network for the given training set. This can produce a localised solution leading to poor performance on the general data set. Secondly, the incremental values in the weight vector may be adjusted in small increments leading to a slow convergence on the solution.

The effects of these problems can be overcome by including a momentum term in the training equation;

$$\text{Momentum Term} = \alpha\left(w_{ji}(t) - w_{ji}(t-1)\right) \ldots\ldots\ldots\ldots \text{eq } 7.17$$

α is the momentum rate and must lie between 0.0 and 1.0. The new weight adjustment function is now

$$w_{ji}(t+1) = w_{ji}(t) + \beta\partial_{pj}o_{pi} + \alpha\left(w_{ji}(t) - w_{ji}(t-1)\right) \ldots \text{eq } 7.18$$

The momentum adds a term proportional to change in weight values. This has the effect of decreasing training times by giving large changes at the start of training giving way to smaller changes as the system converges towards a solution. The large changes at the start of the cycle also helps to 'jolt' the system out of any local minima encountered during the early training phases reducing the chances of a poor solution.

The values of the learning and momentum rates can be adjusted to alter the performance of a network.

7.3.2 Alternative training methods

Many methods have been devised which alter the fashion in which a network is trained. Improvements sought include the speed of training, the accuracy of performance and the prevention of network paralysis. Wasserman [WASSERMANN 1989] and Carpenter [CARPENTER 1989] review some of these methods.

The basic back propagation, three layer network provided adequate performance in the case of shoe component identification. On reviewing the literature describing alternative architectures and methods it was found that although these were generally much more sophisticated they did not promise a vast improvement in performance. These alternative methods were not pursued further.

7.4 Applying neural networks to the classification of shoe components

There are several points which must be addressed when considering the use of neural networks for the identification of shoe components.

- There must be a detectable similarity amongst features representing the same class.
- Sufficient examples of each class must be available for the training of the network.
- The network may take some time to train to a satisfactory level.

One of the original attempts to assess the identification, orientation and position of shoe components employed a neural network system which tried to recognise the components directly from the optical image. The WISARD system [ALEKSANDER 1990 *et. al.*] was a multipurpose neural network designed to identify images of various types. The input to the network was based on samples of the optical image as viewed by the camera, so the inputs were not independent of the components position or orientation. In the case of shoe components the WISARD neural network system was expected to evaluate the components orientation and position in addition to its identity. This meant that the network had to be trained on each shape in a large variety of positions and orientations in order to achieve a reasonable success rate. The variation proved to be too great and the method achieved only limited success.

The moment invariant feature set used to represent the components in this investigation removes the effects of rotation and positional change. If these features are used as inputs to neural networks (which are only used for shape identification) then the network only has to account for variations in input due to noise. This noise is a combination of several factors;

- Quantisation of the image
- Variations in shape features due to pixel arrangement
- Non-unity aspect ratio variations.

The main drawback in component stitching of using a neural network, as with any recognition system where new classes are encountered on-line, is the requirement for a training set. The work of Tout was geared towards a method requiring the presentation of each training shape only once. If neural networks are to be utilised then a training set of several components for each shape will have to be used. The presentation of several shape examples has obvious disadvantages in terms of reduced productivity but may be justified by more accurate recognition.

7.5 Evaluation of neural networks for the classification of shoe components

Using a multi-layer, feed forward network with supervised learning the following factors were considered in order to first establish the suitability of and then to determine the most successful operating conditions for a neural network designed to classify shoe components.

- number of layers
- number of inputs
- number of neurons on each layer
- learning rate
- momentum rate
- acceptable error during training
- iterations in learning cycle
- patterns in training set
- output threshold (for binary decisions)
- conditioning of input vector.

If neural networks are to be used they must successfully discriminate between similar components and be trainable within a production line acceptable time.

7.5.1 Number of layers

The more layers used in a neural network the smoother the class separation boundaries. However the ability of a neural network to separate data classes is not increased by having more layers provided a minimum three layer structure (input, hidden and output layers) is used (Kolmogorov's theorem) [BEALE *et. al.*]. For this reason the architectures used in the following networks are all three layer.

7.5.2 Number of inputs and number of neurons on each layer

Each component is described by a maximum of eight features, F_0 to F_7, giving a network with up to eight input connections. The output is a measure of comparison between the input vector and a taught class, thus a single output neuron is used. The intermediate layer could be any number of neurons but has been limited at the size of the input vector, i.e. eight. Each intermediate neuron can receive any number of the inputs plus one bias value. The basic connectivity framework is illustrated in figure 7.6.

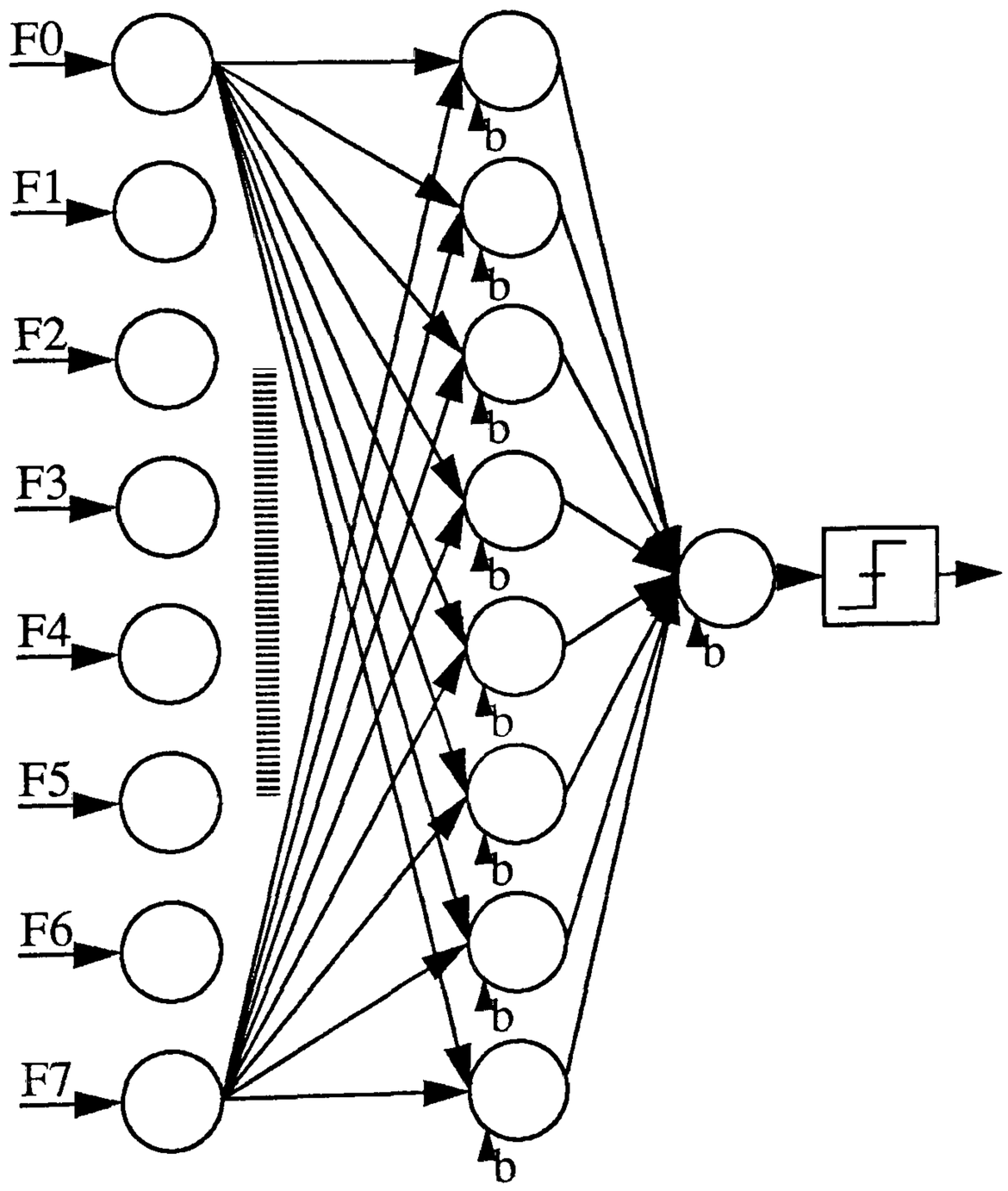
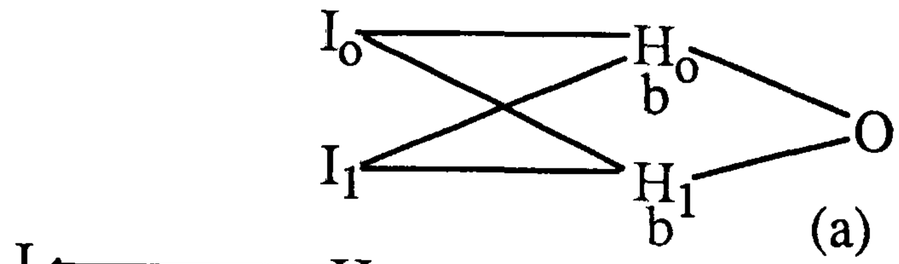
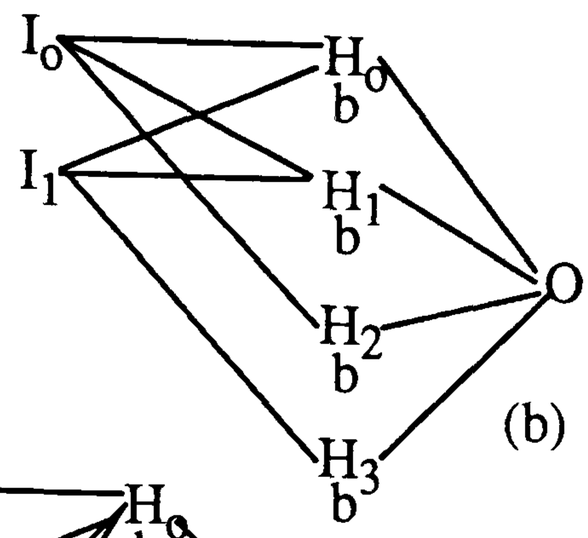
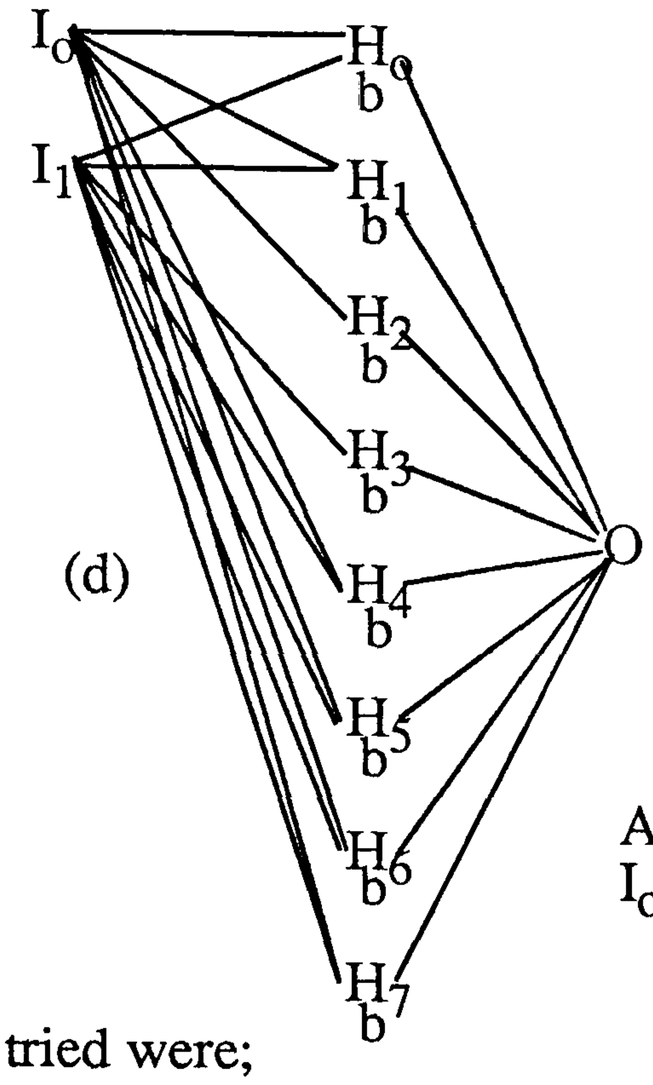


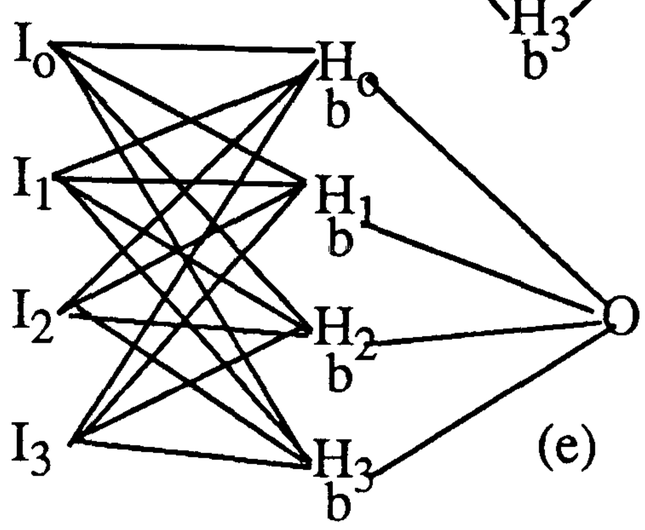
Figure 7.6 The connectivity of the neural network used for each class.

The configurations shown in figure 7.7 were tried.









Above shows
I_o, I_1, I_2, I_3 fully connected to all of H_o, H_1, H_2, H_3.

Also tried were;

(h) I_o, I_1, I_2, I_3, I_4 fully connected to all of H_o, H_1, H_2, H_3, H_4.
(f) I_o, I_1, I_2, I_3, I_4, I_5 fully connected to all of H_o, H_1, H_2, H_3, H_4, H_5.
(g) I_o, I_1, I_2, I_3, I_4, I_5, I_6 fully connected to all of H_o, H_1, H_2, H_3, H_4, H_5, H_6.
(c) I_o, I_1, I_2, I_3, I_4, I_5, I_6, I_7 fully connected to all of H_o, H_1, H_2, H_3, H_4, H_5, H_6, H_7.

Figure 7.7 The network configurations evaluated.

Each of these networks was trained on the same data set containing twelve examples of each of two classes of component. One class was to be associated with an output value of 1.0 (tsh16) and the other with an output value of 0.0 (tsh09). All aspects of the training were identical, the values of the parameters being chosen so as to give a generally smooth training curve converging to zero error:-

Learning rate (β)=0.2

Training iterations=2000

Momentum rate(α)=0.1

Sigmoid function spread (k)=1.0

The input vector features were scaled:- F0 by $1x10^4$, F1 by $1x10^8$, F2 by $1x10^{19}$, F3 by $1x10^{25}$, F4 by $1x10^{24}$, F5 by $1x10^{49}$, F6 by $1x10^{34}$, F7 by $1x10^{48}$.

After the training phase 36 different examples of each class (not contained in the training set) were passed through each network and the corresponding output values

noted. Figure 7.8 (a to h) show the network output for each sample. The true class (tsh16) indicated by squares should approach 1.0 while the false class (tsh09) indicated by the triangles should approach 0.0. Shown next to each output graph is the effect on the substitution and rejection rates of a decision threshold evaluated between 0.0 and 1.0.

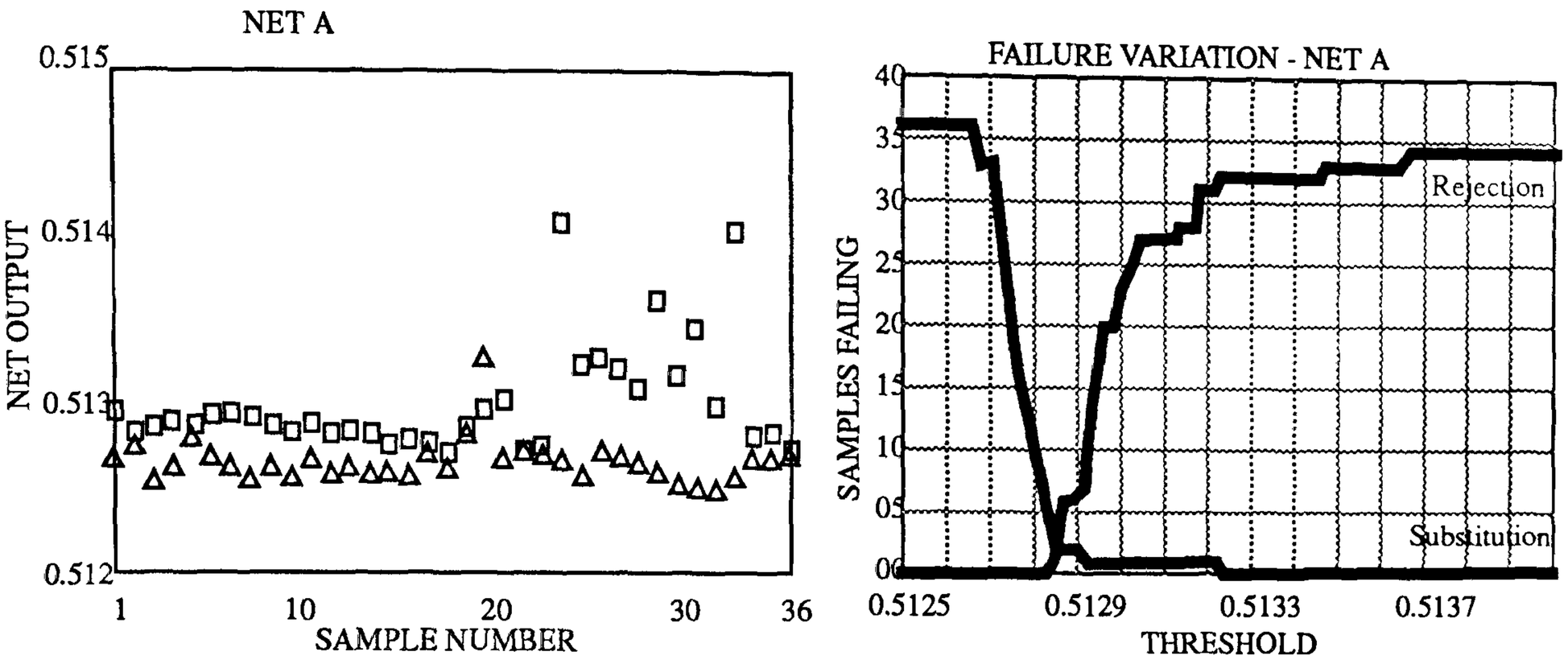


Figure 7.8 (a).

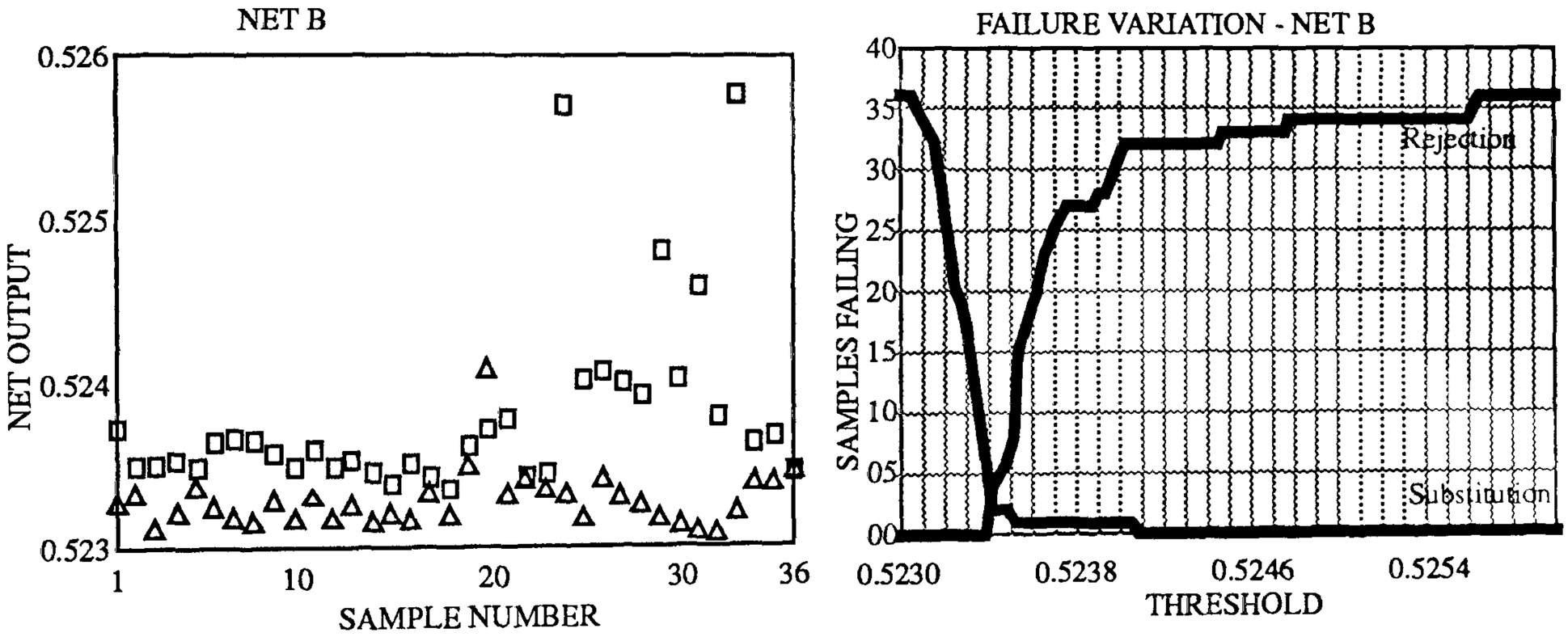


Figure 7.8 (b).

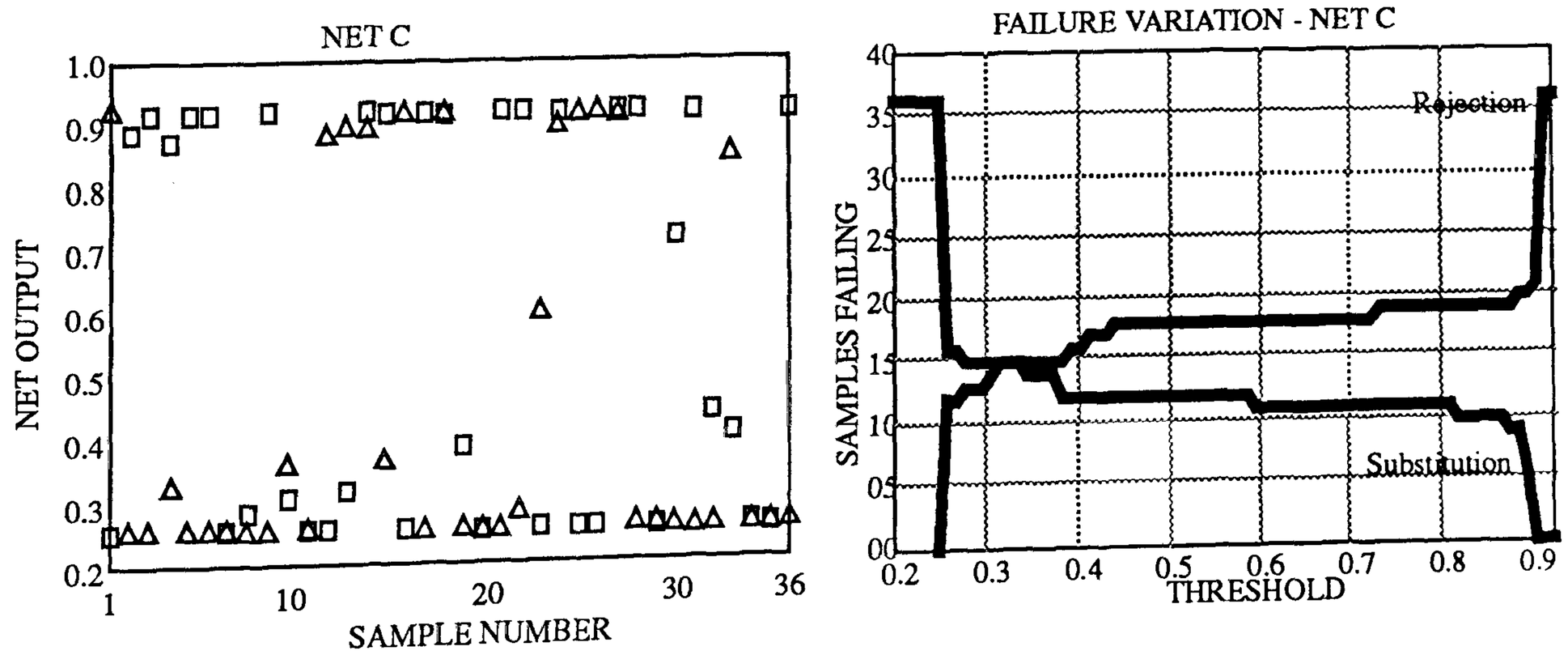


Figure 7.8 (c).

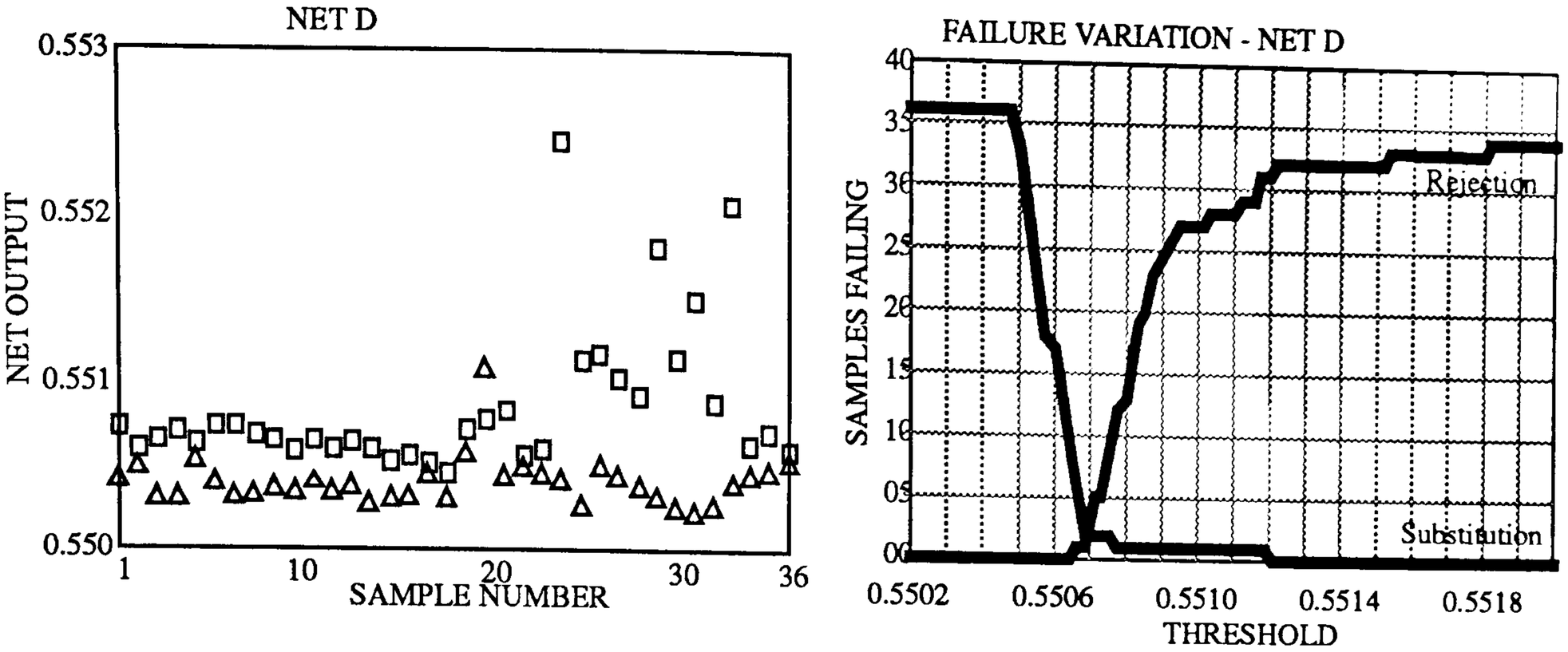


Figure 7.8 (d).

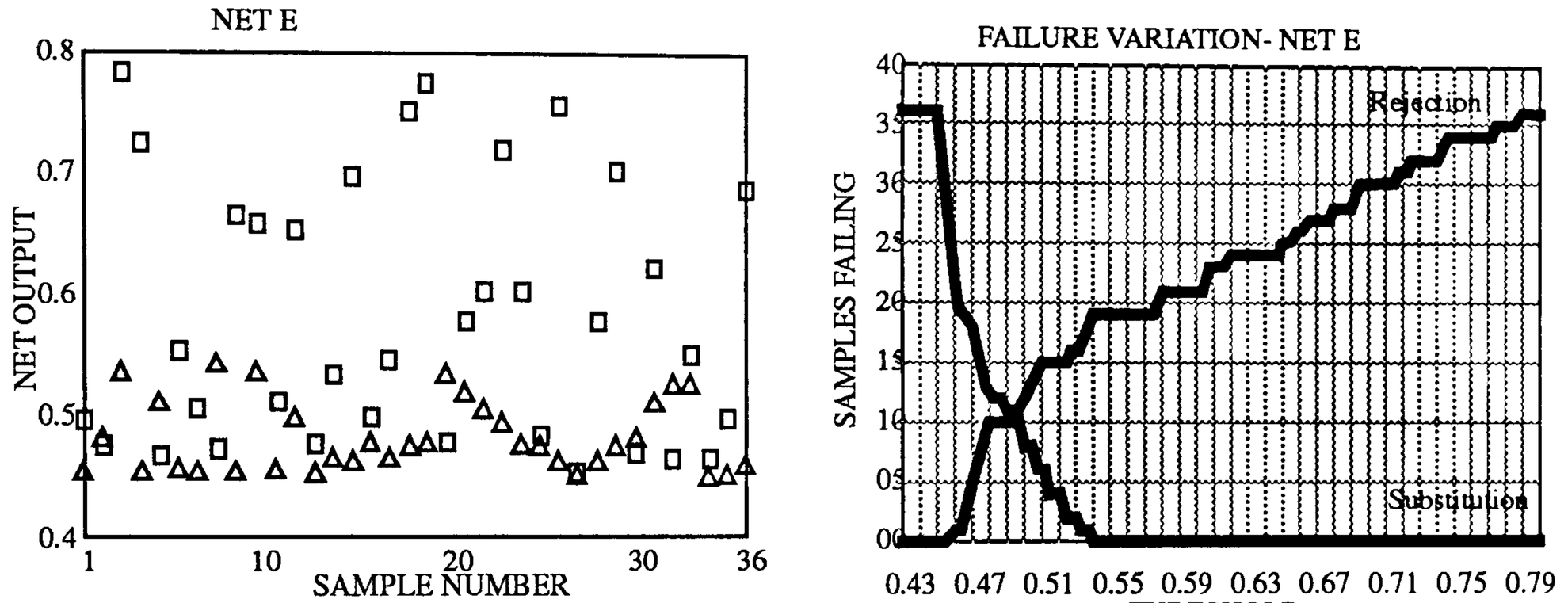


Figure 7.8 (e).

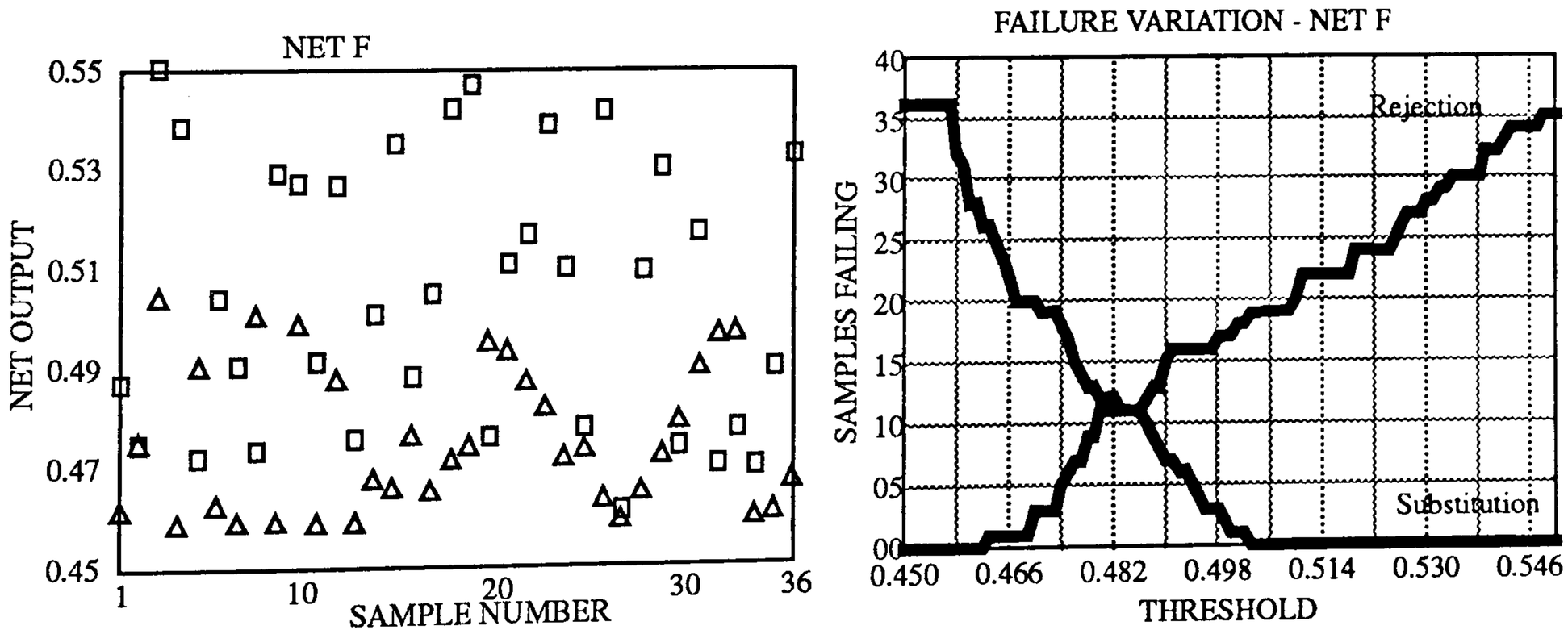


Figure 7.8 (f).

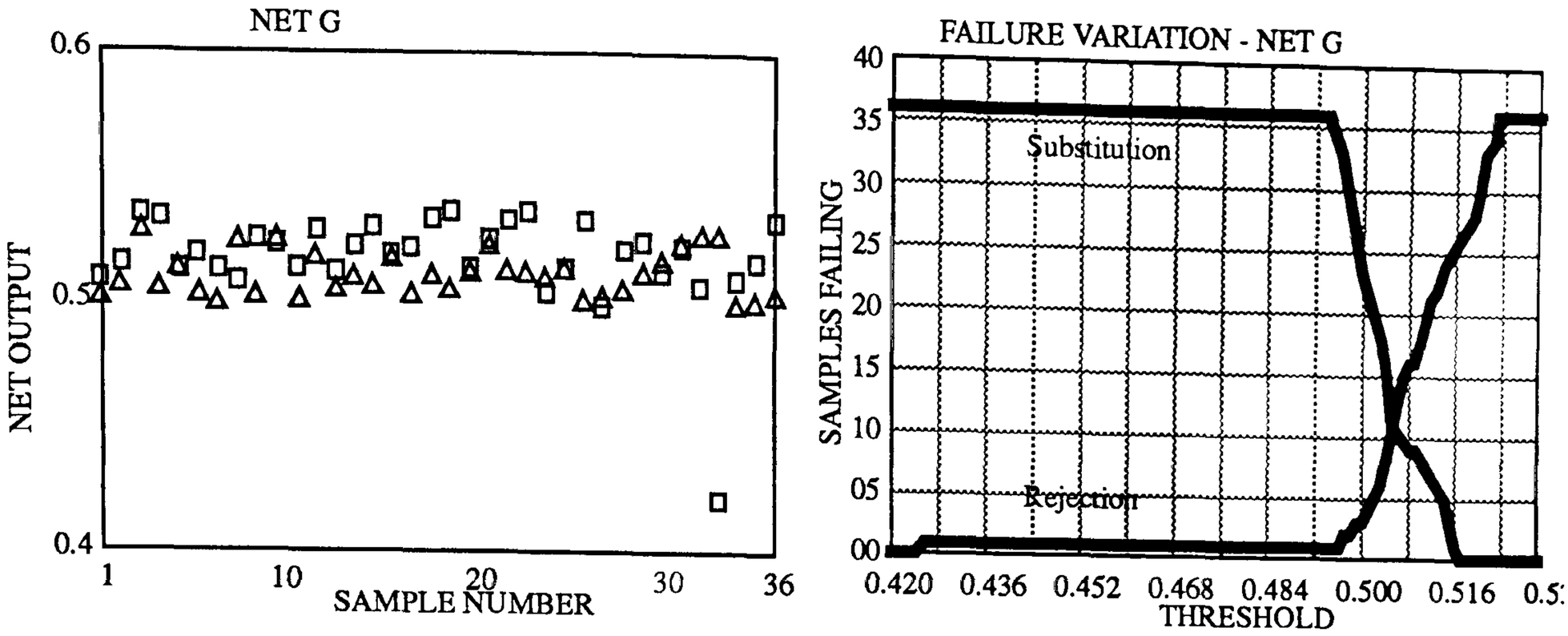


Figure 7.8 (g).

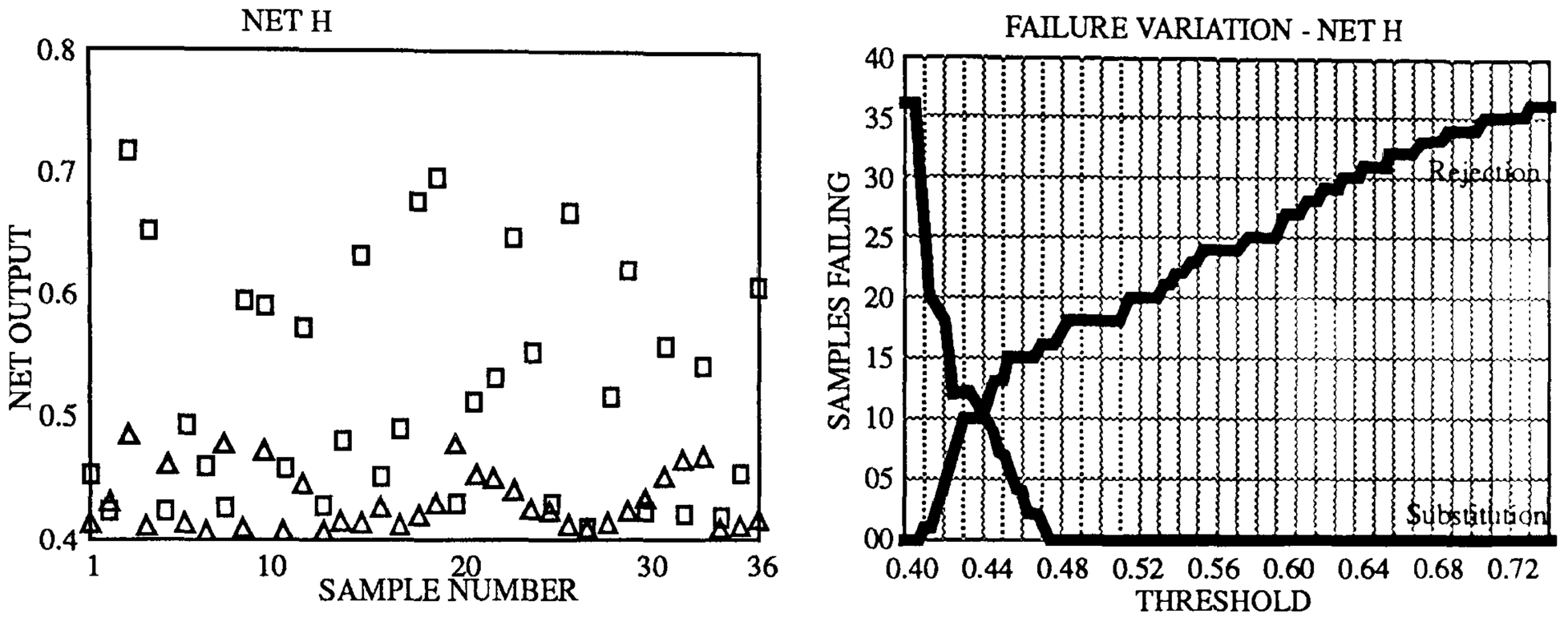


Figure 7.8 (h).

These results were not very encouraging and required significant improvement in the segregation ability of the networks if they were to be of any use.

7.5.3 Conditioning of input vector

An improvement in performance can be obtained if the input data vector is shifted so that certain features of the true class are moved towards the origin. The alterations to the input vector are selected during training. The first example of the true class is taken as the shift vector. When the shift vector is applied to this first example of the true class it is moved towards the origin. All the remaining examples of the true class are moved closer to the origin while the false class vectors are shifted to points more distant from the origin.

In the experiment the feature vectors were scaled as before and then features F0 and F1 were shifted (subtraction) by 41.84890 and 364.3398 which were the F0 and F1 values for the first example of the true class.

The results of shifting the data can be seen in figure 7.9 (a to h).

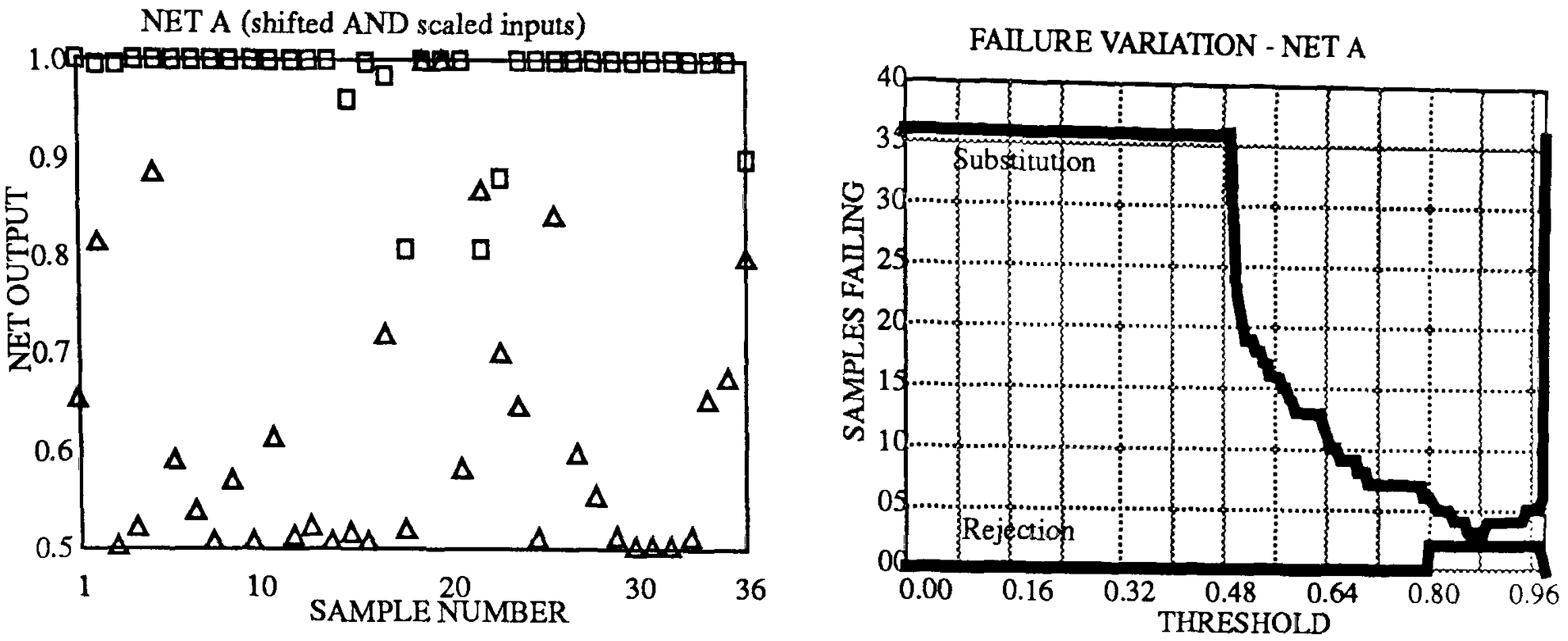


Figure 7.9 (a).

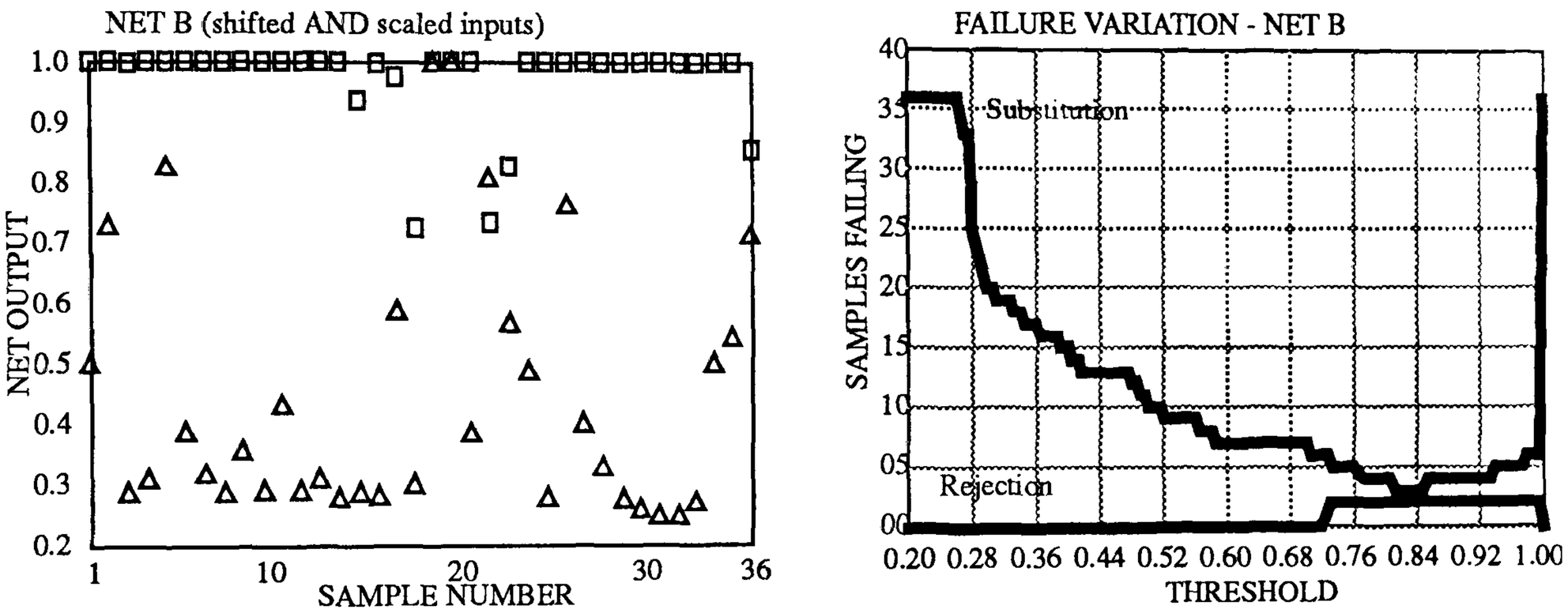


Figure 7.9 (b).

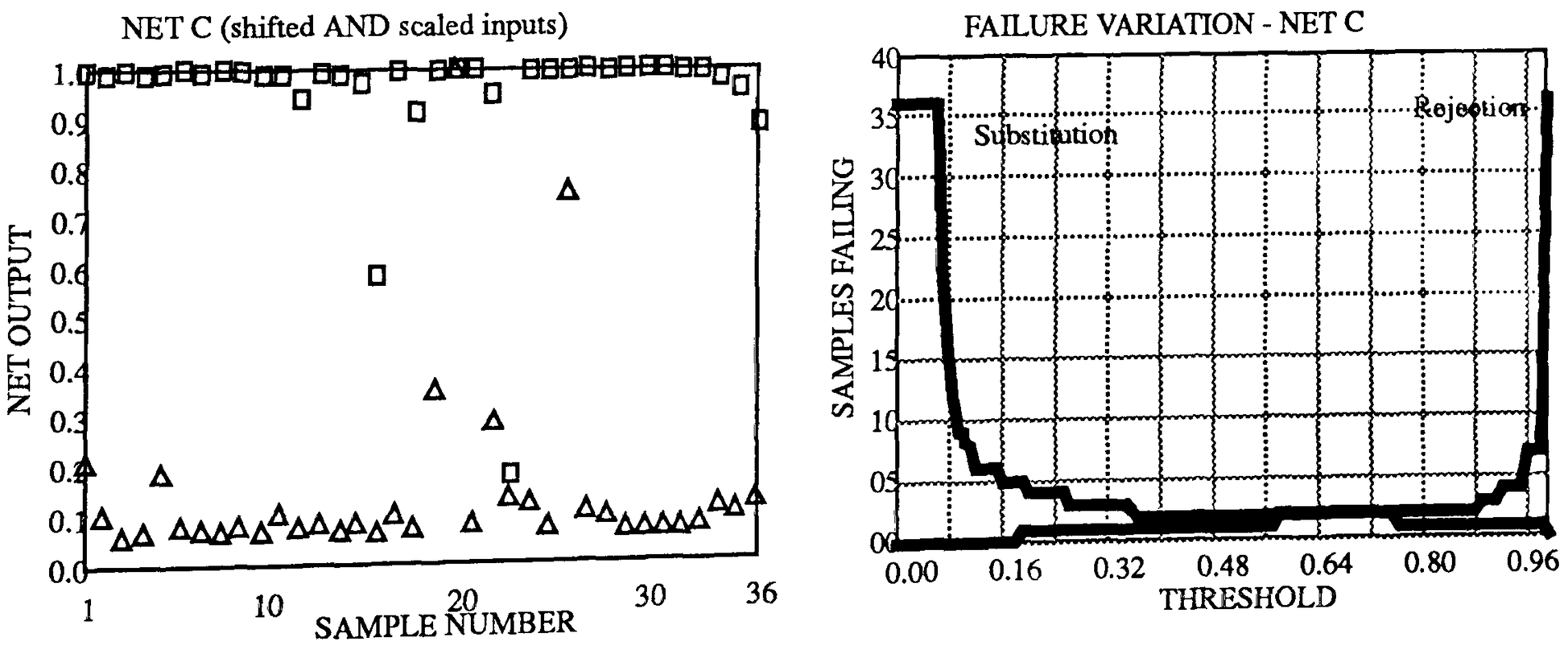


Figure 7.9 (c).

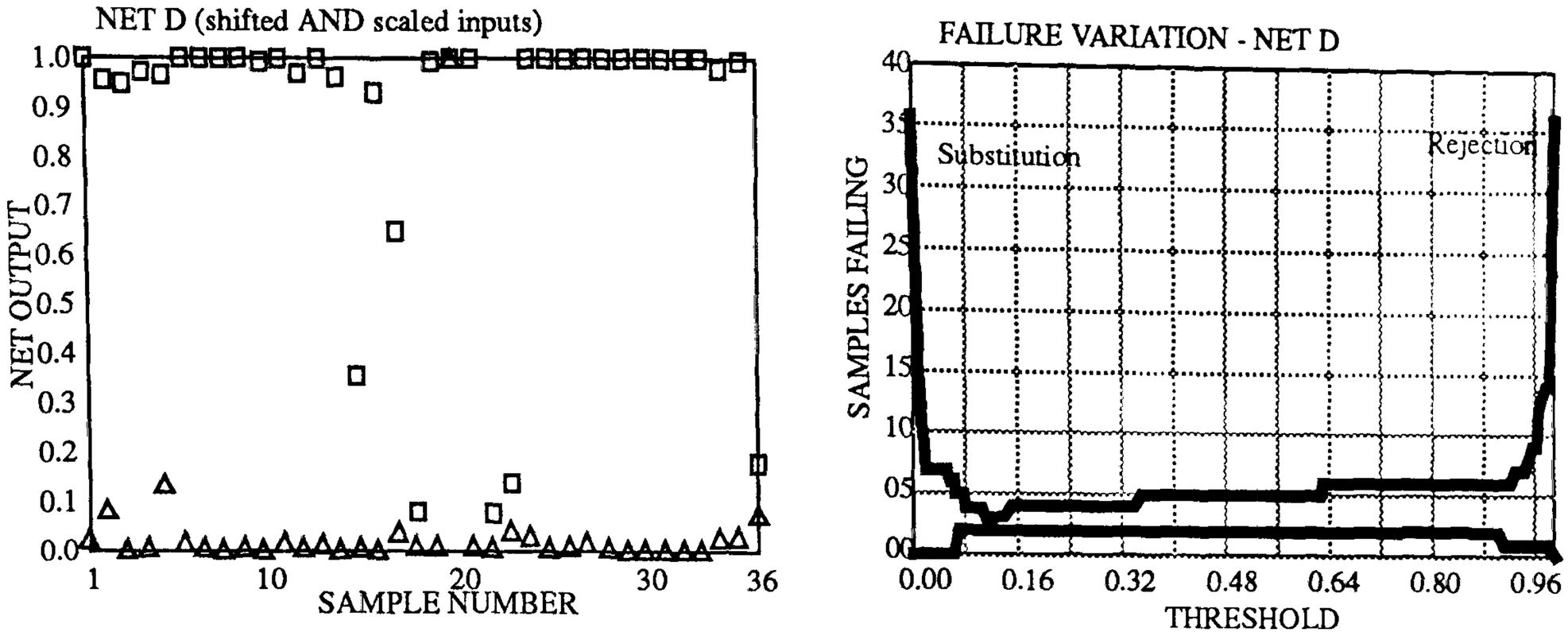


Figure 7.9 (d).

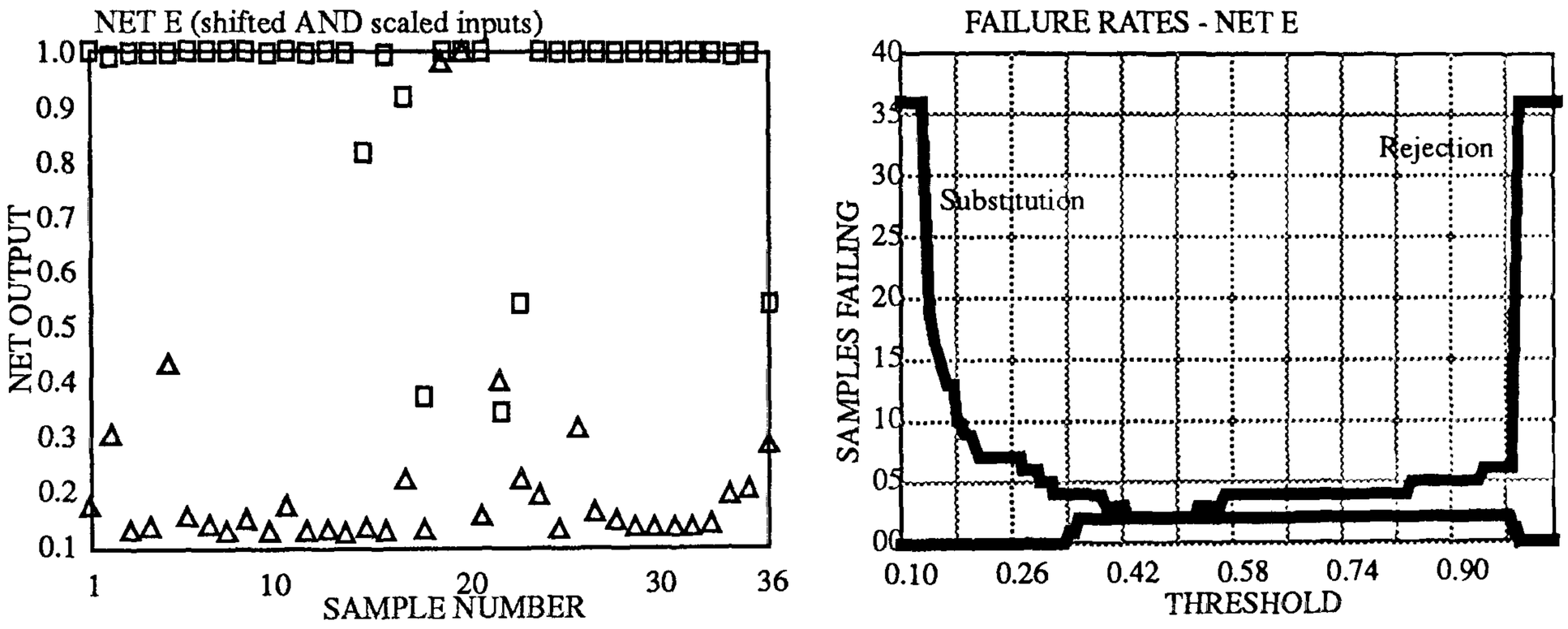


Figure 7.9 (e).

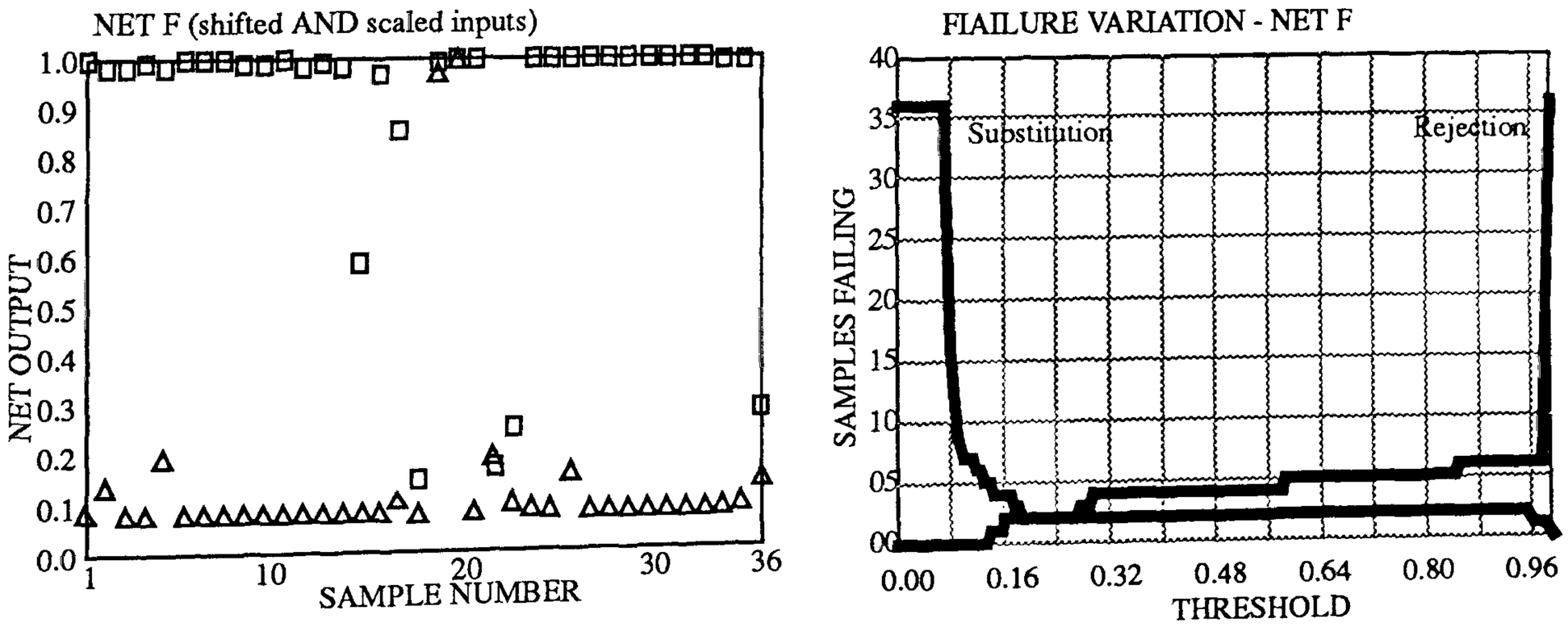


Figure 7.9 (f).

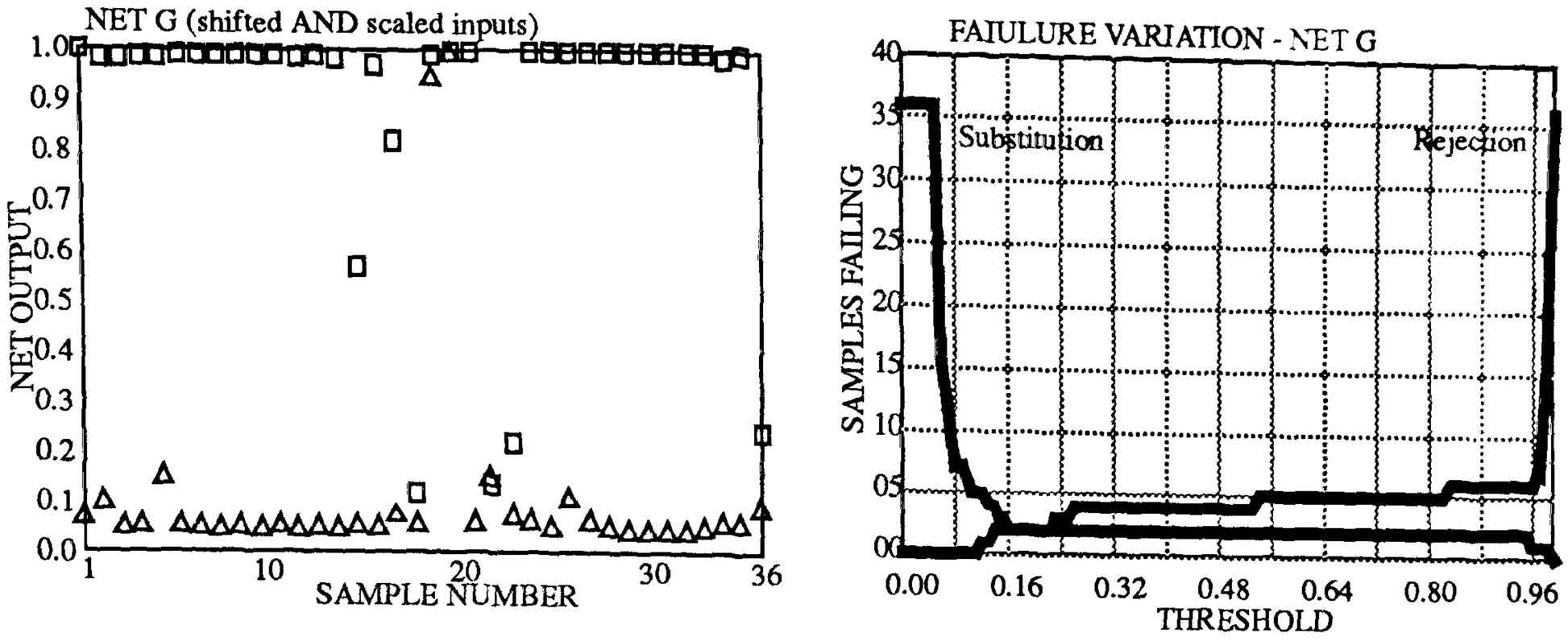


Figure 7.9 (g).

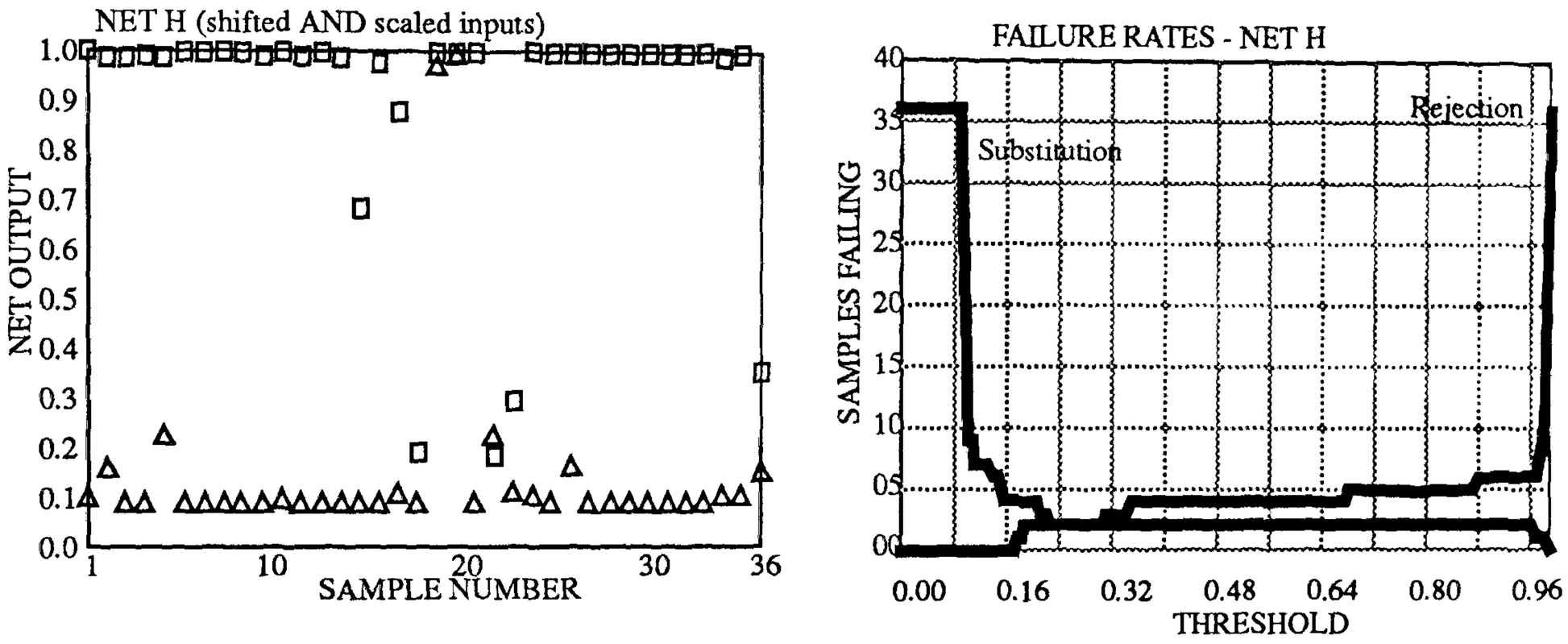


Figure 7.9 (h).

By shifting the input vector the performances of the neural networks are improved. The separation of the two classes is increased and the failure rates due to rejection and substitution are reduced to lower levels over a greater range of the threshold variable. The selection of a threshold value is now less critical and more likely to be generally suitable.

The best architecture is C with all 8 inputs features connected to eight hidden nodes and one output node. The eight hidden nodes are also individually connected to a bias input. The selected output threshold is 0.5 to give a general purpose network.

7.5.4 Learning and momentum rates

Varying the learning rate β and momentum rate α of the training equation will change the speed and smoothness of convergence for a given training set and also the performance for a given general data set. The number and variety of shoe components

likely to be encountered in a shoe factory means that any values of α and β must be chosen so as to give a general performance which is fast and accurate enough to be of use in an on-line training system using unknown components. Figure 7.10 (a to h) show how the variations of the learning rate and the momentum rate affect the convergence time. The convergence time is taken as the number of iterations required before the error has fallen below 0.001 for 50 consecutive passes.

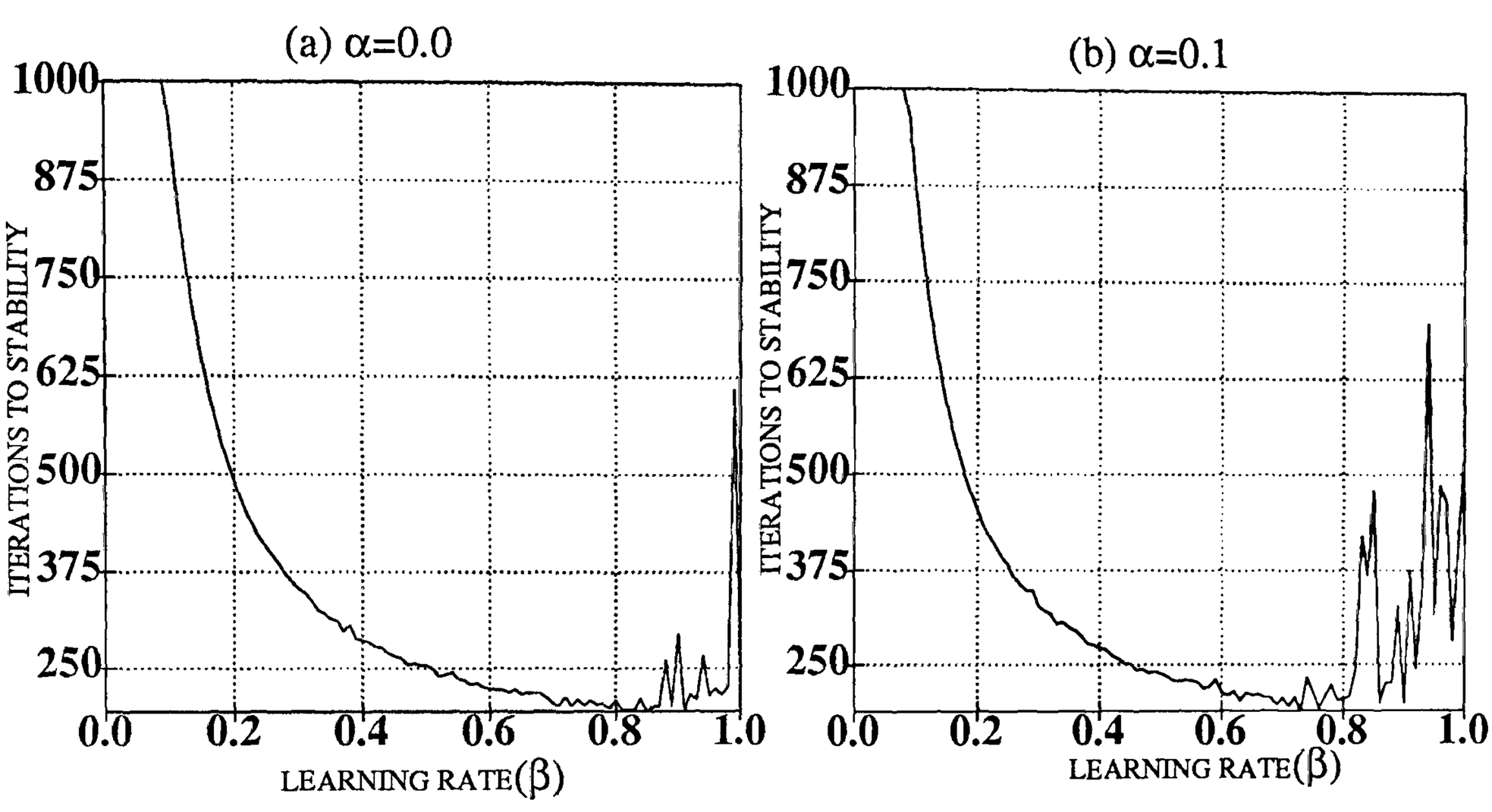


Figure 7.10 Iterations to stability for different parameters.

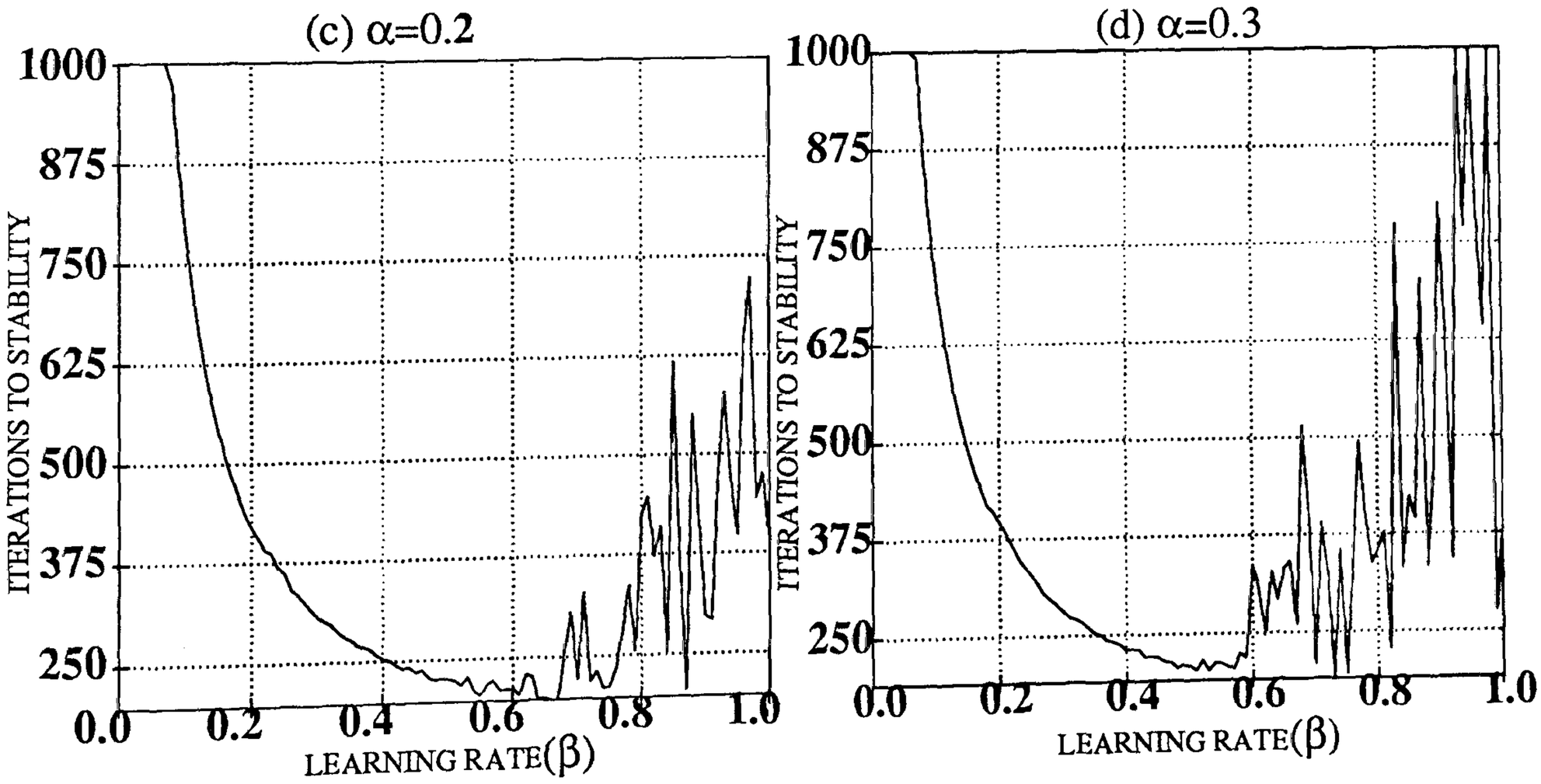


Figure 7.10 Iterations to stability for different parameters.

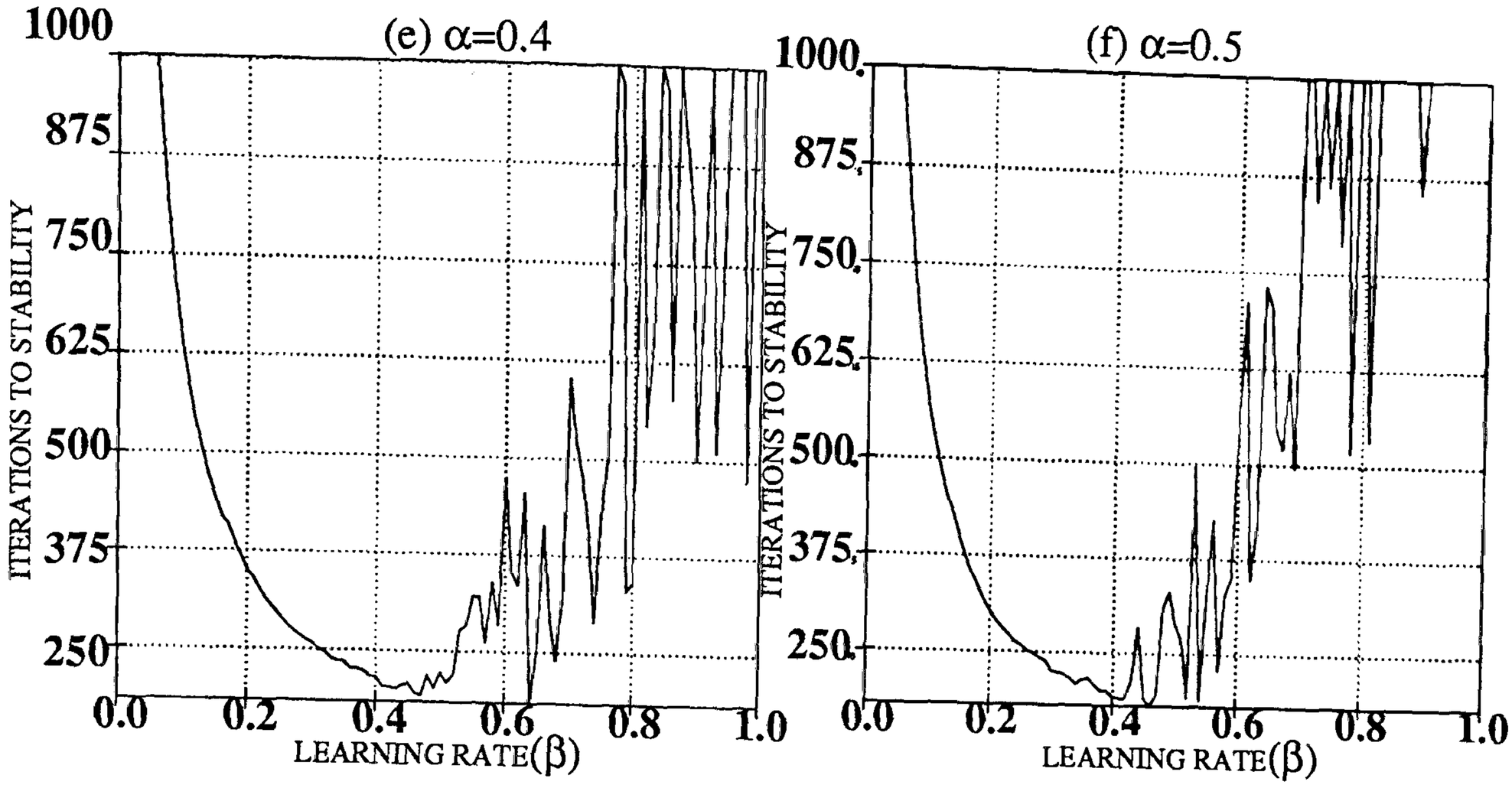


Figure 7.10 Iterations to stability for different parameters.

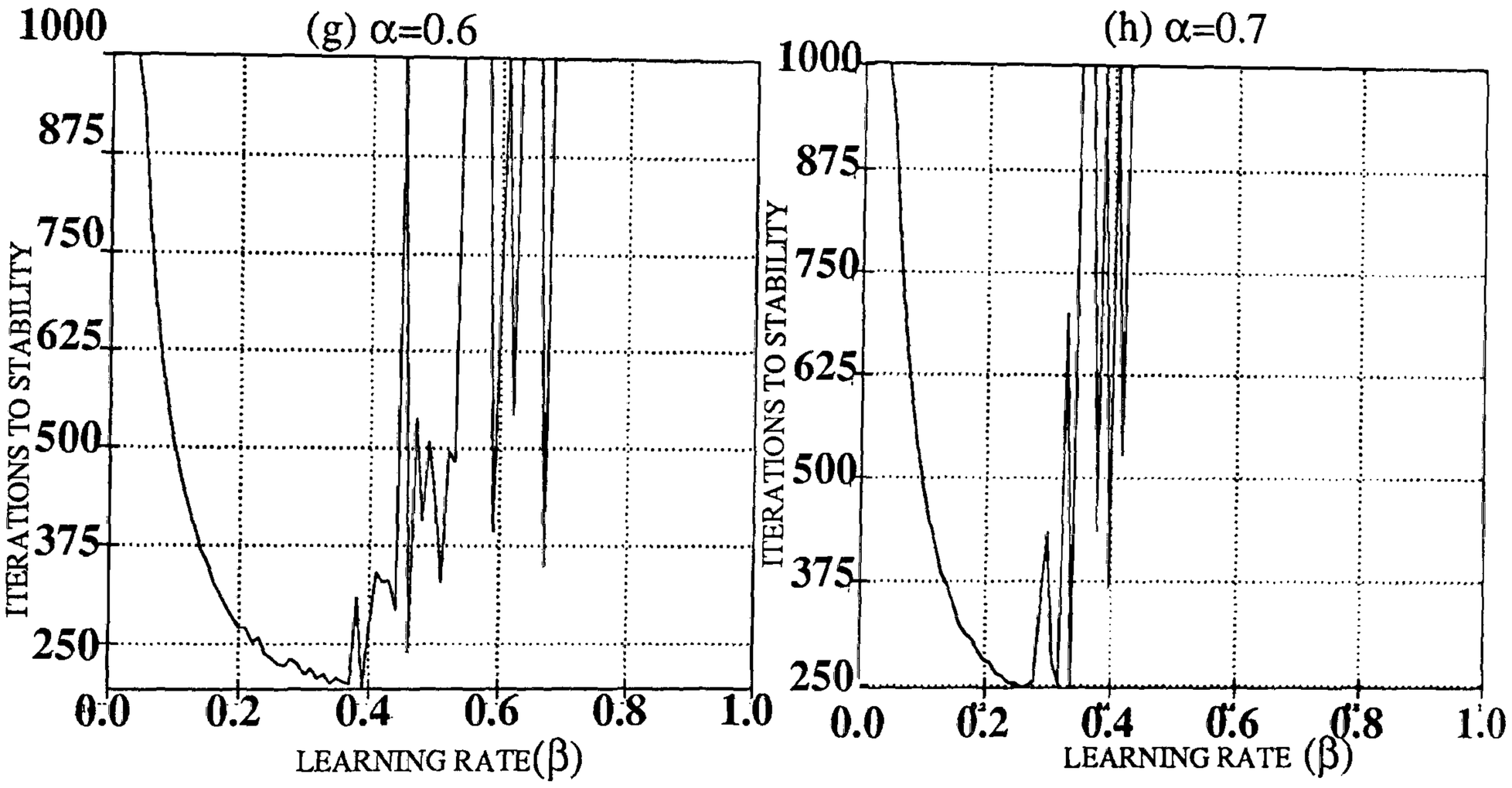


Figure 7.10 Iterations to stability for different parameters.

In general an increase in the learning rate produces a decrease in the training time. Similarly if the momentum rate is increased then the training time for a given learning rate decreases. For large values of β (learning rate) this relationship breaks down and the required number of iterations becomes unpredictable. The critical value of β decreases as α (momentum rate) increases. This relationship is shown in figure 7.11.

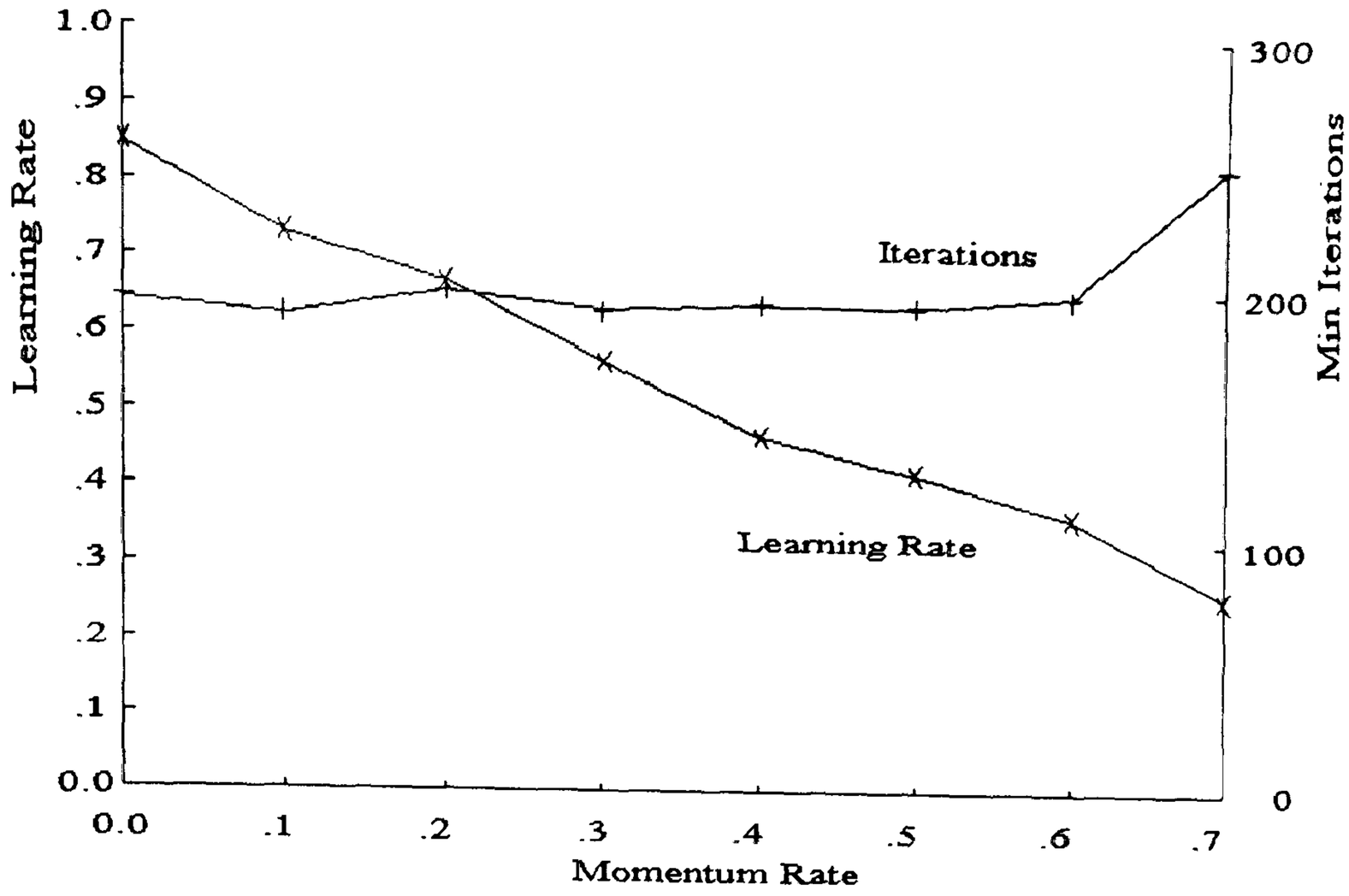


Figure 7.11 Summary of figure 7.10 at stability breakdown.

Figure 7.11 also shows the minimum number of iterations for each value of α. The minimum number of training passes needed to achieve the required error is consistently about 200 iterations.

7.5.5 Iterations in learning cycle

The lower the error value reached during the training phase the more successful the recognition performance of the network. The level of the error is generally reduced with further iterations of the training algorithm. The training cycle can be stopped either when the error reaches a certain level or after a given number of training cycles. Provided a training error of around 0.001 was achieved the network used could distinguish the shapes presented before it. The number of iterations taken to achieve this error value varies with the network parameters but must be kept realistic for industrial application purposes. If a learning rate of 0.40 and a momentum rate of 0.2 is used then figure 7.10 (c) shows that the number of iterations will be consistently around 250 for the data sets used for the investigations. The choice of momentum rate is important since too large a value can lead to a network where the convergence time is wildly unpredictable, too small a value and the network may become stuck in a local minimum.

7.5.6 Patterns in training set

A neural network requires many examples of each class if it is to achieve sufficient training. The presentation of each training example consumes production time and so it is important to minimise the required number of class examples.

Table 7.1 shows a comparison between the number of class examples used for training purposes and the network performance in separating classes Tsh16 and Tsh09. The number of training patterns used will depend on the required system performance.

Training examples per class	Average number of recognition errors in 82 test shapes
1	23.75
2	9.25
3	7.50
4	8.25
5	7.25
6	6.75
7	5.00
8	4.80
9	4.30
10	4.00
11	3.50
12	3.10
13	3.00
14	3.00

Table 7.1 The effect of training set size on recognition performance.

7.5.7 Output threshold (for binary decisions)

The neural network is required to make a yes/no decision regarding the identity of a component. The network is configured in a unipolar fashion so its output is in the range 0.0 to 1.0. Ideally an incorrect component will produce a value approaching 0.0 while a matching component will produce a value nearer to 1.0. In this ideal case a threshold value can be set at 0.5 such that all outputs below 0.5 produce a 'no' output and all values above 0.5 produce a 'yes' output. In reality it was found that two very similar classes both produced output levels approaching 1.0 but still distinguishable provided a suitable threshold level was chosen. The choice of threshold is best performed during the training of the network when the average output of each class can be monitored and the threshold chosen as the value halfway between them.

7.5.8 Alternatives to manual presentation of training data

The main drawback envisaged in the use of neural networks for shoe manufacture is the high workload required by network training. If it takes 10 seconds to present each shape and 10 examples of each class are required then a database of 20,000 shapes would require 556 hours to input as opposed to 56 hours for a single example system.

The viability of such a complete system would depend on being able to automatically generate the training sets from a single example. Physically this might be achievable via a robot arm or air jets but it is more feasible (and faster) if tackled in software:

The problem of shoe component identification is theoretically deterministic, with all components of the same class being identical. In practice this is not the case and the fundamental cause of identification failure is the lack of a perfect prototype which is noise free. As part of a 'complete manufacturing system' each stitching machine could have access to the original CAD shoe design allowing a perfect prototype to be obtained from which a training set could be computer generated. It is, however, likely that the feature matching approached used by Tout and proposed by Norton-Wayne [Norton-Wayne 1982] would also show a considerable improvement in performance given such an ideal prototype.

Following on from the computer generation of a training set, experiments were performed using a single shape example as the basis for an artificially generated training set of 20 examples. The recognition performance depended on the relationship between the prototype and the class mean. Only those prototypes with very similar features to the class mean produced a trained network with adequate performance.

7.5.9 A single network approach

As already stated the main problem with a neural network approach to an open ended problem such as industrial shoe component identification is the continuous network training. In the approach described previously a network is trained to give a high response to a 'true' class and low responses to other classes. This method trains one network for each class as it is encountered and thus requires as many networks as classes.

A global approach was investigated where a single network was trained to recognise prototype/component pairs of feature vectors from examples of the same class and reject pairs from differing classes. The network had sixteen inputs corresponding to the features of the prototype and the unknown component. During training an output of 1.0 corresponded to a matching pair and an output of 0.0 to a non matching pair. The single network required for all recognition was trained extensively (300 component pairs and 2,000 passes) but proved unsuccessful due to the differing relationships between features for different classes.

Compensating for imaging grid errors as described in section 6.6.2 may provide an improvement to the results for the single neural network method by reducing the feature variation between 'identical' components.

7.6 Summary

The experiments demonstrated that neural networks could be used to separate even the closest of shoe components. The requirement for training, using examples of each shape, is a drawback for any production line system and makes the complete use of individual networks unlikely for a 20,000 component system.

A single network trained to recognise the similarity between identical components rather than the components themselves would be ideal since training time prior to system use would not be a problem. The results for such a method were not encouraging.

Overall neural networks are good shoe component discriminators which must be used in moderation to avoid overburdening the production line. Chapter 8 shows how neural networks might be utilised in a composite system by limiting their use to difficult identification problems.

Parameter	value
number of layers	3
number of inputs (nodes)	8 + bias
number of hidden nodes	8
number of output nodes	1
connectivity	pattern c (section 7.5.2)
learning rate	0.4
momentum rate	0.2
max learning iterations	250
training set size	6
training error target	0.001
binary threshold level	mid point of training values
threshold type	sigmoid function

Table 7.2 The neural network parameters chosen for on-line use.

Table 7.2 shows the parameter values selected for the neural networks used in chapter 8. The neural networks are used only for identifying groups of components which can not be identified using the feature comparison approach discussed in chapter 4.

CHAPTER 8

PRACTICAL IMPLEMENTATION AND SYSTEM PERFORMANCE

8.1 Introduction

Previous chapters have shown that shoe component shapes can be represented effectively by features based on moment invariants and that the position and orientation of the components can be obtained from the moment values. The identification of the components can be achieved through the use of neural networks but these are generally cumbersome and time consuming to train. The feature comparison method used by Tout requires only one training example per class but is more fallible than neural networks due to the effects of noise on the moment values.

The following sections describe a hybrid method making use of both neural networks and feature comparison and compares the results with those achieved by Tout's area-radii-second moments system.

8.2 Overview

As each image is scanned by the camera it is filtered (section 6.6.1), converted into its moment form and the real image is discarded. The moments are compensated for grid errors (section 6.6.2) and then used to calculate the moment invariants, component position and orientation. The actual processing of the component will depend on whether the system is in *learn* or *recognition* mode and the GCOEFF value of the component.

In *learn* mode all classes are stored as a single prototype of eight features (F0 to F7). If a component is presented for training which is within a certain tolerance of an existing prototype then further examples of each class can be gathered and a neural network trained to separate the two or more similar classes.

During the *recognition* phase components are compared on a feature matching basis with the stored prototypes. If a match is found for which a neural network has been

trained then the necessary networks for that component group are run to identify the component. When two or more prototypes are candidates and a neural network has not been trained then identification is resolved by finding the prototype with the highest number of nearest features. This requires the calculation of the one dimensional Euclidean distance for each feature for each prototype from the unidentified component.

8.2.1 Training scheme

The new components are presented to the system. The moment invariant features (F0 to F7) are extracted; the centroid is located (eq 6.3, eq 6.4) and the value of GCOEFF (eq 6.8, eq 6.9, eq 6.10) is calculated.

If GCOEFF is greater than 0.2 the orientation of the shape can be calculated using equation 3.6. This value will be stored along with the eight moment invariants describing the shape and the coordinates of its centroid. For those shapes with GCOEFF less than (or equal to) 0.2 the values of μ_{30}, μ_{03}, $\mu_{21}$and μ_{12} are stored in place of the orientation value along with an indication of the shapes' low GCOEFF value.

8.2.1.1 Using neural networks

The moment invariants describing the shape prototype are compared with those already present in the data base using a feature matching approach. If any other shape prototypes match the new shape within a certain tolerance (i.e. fall within the same group of component shapes) then additional images of each component are gathered.

Neural networks are taught to separate these class groups with one network for each prototype. One network in each group is taught to respond to each class while rejecting all others. The neural network parameters are stored along with the classes which they have been taught to separate. The use of many networks as opposed to one network with many output avoids the need to limit the systems capabilities.

8.2.1.2 Component processing

In the case of a component stitching system the component identification, position and orientation are passed to the sewing machine module which records all the data points as the prototype pattern is stitched using a joystick unit.

Finally the component is ejected and the system awaits the next component.

8.2.2 Recognition process

As with the training scheme the component is presented to the system and the moment values used to calculate the moment invariants and the position of the components centroid.

8.2.2.1 Primary classification by Hamming distance measure

The system compares the eight features measured with those stored for each shape in the database. The method of comparison is that based on the Hamming distance measure [NORTON-WAYNE1982] described in section 4.3 using the tolerances given in table 8.1. If the data base search can not find a component prototype that matches the new shape with a minimum number of features then it will reject it and the operator will either re-present the component or discard it as faulty. Provided that only one prototype matches the component in sufficient features and that it has not been used to train a neural network then that prototypes identity is used to select appropriate operations for the component.

Feature	Tolerance (%)
F0	0.3
F1	1.5
F2	3.0
F3	4.0
F4	6.0
F5	20.0
F6	30.0
F7	+ve or -ve
Minimum Match	5 features from 8

Table 8.1 Tolerances for feature matching

8.2.2.2 Component orientation

If the data base indicates that the prototype has a low GCOEFF the values of μ_{30}, μ_{03}, $\mu_{21}$and μ_{12} are extracted, combined with the values calculated for the new shape and used to evaluate the orientation as described in section 6.4.3.2 by finding the solution to

equation 6.16. If GCOEFF is greater than 0.2 the orientation of the shape can be calculated using Equation 3.6.

8.2.2.3 Secondary classification by neural networks

If the primary classification identifies a component as being of a class contained in one of the groups used for network training then those networks are invoked and used to re-identify the component. Similarly if two or more prototypes match the same number of features then any networks trained on the relevant prototypes are used. In order to achieve identification the output of the networks must be evaluated:

Two methods have been tried. The first applies a threshold to the network outputs giving a binary decision as to whether or not the network has fired and the component matches the corresponding class. This can produce further ambiguity since more than one of the networks in the group may fire, producing several further possibilities as to the components identity.

The second method removes the output threshold function and uses the activation value of the output neuron as a value indicating how closely a component is to the trained prototype. The value obtained must be adjusted to take account of the distribution of output values encountered during the training of all the networks in the group. This is achieved by finding the average output value for each network for the training set and offsetting the actual output by this amount. A network giving outputs of 0.9 and 0.3 for true and false classes respectively thus produces outputs of 0.3 and -0.3. The sensitivity of the output is then adjusted so that all outputs are over the range -1.0 to +1.0. An unknown component is assigned to the prototype associated with the neural network producing the highest output value. This method provided the better performance and is used in the experiments detailed in section 8.3.

Assigning a component simply on the basis of the highest value means that no component can be rejected (by the neural network stage) and can lead to substitution if two (or more) networks produce similar output values. The relative error rates can be adjusted by stipulating that components are only assigned if the corresponding neural network outputs are above a certain value and/or exceed other network outputs by a given amount.

If several prototypes have the same number of matching features but have not been used to train a network group then identification is resolved by comparing the prototypes with the unknown component on a feature by feature basis. For each feature the closest prototype scores '1 vote'. The shape is assigned to the prototype with the

highest number of votes. If prototypes have equivalent votes then the component is assigned to the first prototype.

8.2.2.4 Component processing

The component identification, position and orientation can then be passed to any subsequent processes. In the case of component stitching this would be a sewing machine module which would select the appropriate pattern based on component identity. The pattern is adjusted for the new shape position and rotated using equations 3.8 & 3.9. The sewing machine module then controls the component rollers and sewing machine while the pattern is stitched.

Finally the component is ejected and the system awaits the next component.

8.3 Recognition performance

Using the test set given in appendix IV the average results for the composite system are given in table 8.2.

TRUE SHAPE \ ALLOCATED SHAPE	Tsh01	Tsh02	Tsh03	Tsh05	Tsh06	Tsh08	Tsh09	Tsh10	Tsh11	Tsh12	Tsh13	Tsh14	Tsh15	Tsh16	Tsh19	Tsh20	Tsh21	Tsh23	Tsh28	Tsh38	REJECT
Tsh01	100																				
Tsh02		100																			
Tsh03			93.9																6.1		
Tsh05				100																	
Tsh06					96.1																3.9
Tsh08						97.2															2.8
Tsh09							87.2							12.8							
Tsh10								100													
Tsh11									94.5	4.4	1.1										
Tsh12									2.2	97.8											
Tsh13									2.5		97.5										
Tsh14												100									
Tsh15									0.6				99.4								
Tsh16							11.7	2.2						83.1	3.0						
Tsh19															100						
Tsh20																97.2					2.8
Tsh21																1.1	86.6	1.7			10.6
Tsh23																0.6	0.6	98.8			
Tsh28			14.4																80.0		5.6
Tsh38																				100	

Table 8.2 Average recognition results using feature matching and neural nets.

To summarise;-

The average substitution rate (components allocated to wrong class) is 3.25%

The average rejection rate (components unidentified) is 1.29%

Average total failure rate (sum of substitution and rejection errors) is 4.54%

The summary results are obtained by adding all the (e.g. substitutions) together, dividing by 2000 (the total number of tests) and multiplying by 100%.

Neural networks were used to separate the following groups of components; tsh09 and tsh16; tsh20, tsh21 and tsh23; tsh03 and tsh28; tsh11, tsh12 and tsh14. Six examples of each prototype were used for training the networks.

These results are an improvement on those for moment invariant feature identification using feature matching only (table 6.3). The results using the polar coding approach are presented in table 8.3.

TRUE SHAPE \ ALLOCATED SHAPE	Tsh01	Tsh02	Tsh03	Tsh05	Tsh06	Tsh08	Tsh09	Tsh10	Tsh11	Tsh12	Tsh13	Tsh14	Tsh15	Tsh16	Tsh19	Tsh20	Tsh21	Tsh23	Tsh28	Tsh38	REJECT
Tsh01	100																				
Tsh02		100																			
Tsh03			65.6																34.4		
Tsh05				100																	
Tsh06					100																
Tsh08						100															
Tsh09							85.0							15.0							
Tsh10								100													
Tsh11									85.0		15.0										
Tsh12										86.7			13.3								
Tsh13									0.6		90.4										
Tsh14										6.7		93.3									
Tsh15										2.2			97.8								
Tsh16							6.7							93.3							
Tsh19															100						
Tsh20																100					
Tsh21																	95.6	4.4			
Tsh23																	2.8	97.2			
Tsh28			33.9																66.1		
Tsh38																				100	

Table 8.3 Average results for polar vector identification.

To summarise;-

The average Substitution Rate is 6.75%

The average Rejection Rate is 0.00%

Average total failure rate is 6.75%

8.3.1 A comparison of performances

	Substitution rate	Rejection rate	Total Fail Rate
Unfiltered shapes (table 6.2)	6.44%	1.56%	8.00%
Filtered shapes (table 6.3)	6.20%	1.83%	8.03%
Adjusted features (table 6.4)	4.86%	1.29%	6.15%
Neural networks (table 8.2)	3.25%	1.29%	4.54%
Polar vector (table 8.3)	6.75%	0.00%	6.75%

Table 8.4 Recognition performance summary.

Table 8.4 gathers together the results for the four moment invariant methods and the polar vector method. All the moment invariant methods have better substitution rates than the polar coding approach, but the latter method is superior in terms of rejection. The relative proportions of rejection and substitution can be varied by adjusting the feature matching tolerances so it is the overall failure rate which forms the best assessment of performance.

The neural network approach produced the lowest failure rate but has the training time disadvantages discussed in chapter 7. The result could be further improved if more examples were used during network training. Adjusting the moment features to compensate for the imaging grid produces better results than the polar vector method but 0.6% may not be significant in terms of overall performance. An untried system utilising both polar vectors and moment invariants may yet prove to be the best answer.

8.4 Reasons for a high substitution rate

The original specification from BUSM called for a rejection rate of 1 in 10,000 and a substitution rate of 1 in 100,000 over the whole universe of shapes.

The sets of shapes used to test machines for use in areas such as shoe manufacture, where the data set limits are not defined, can only ever be a representative sample of the expected universe. The actual shapes encountered will depend on the whim of the shoe designer so no system can be equipped to cope with every possibility. The shapes used in the testing of this system are a cross section of the types of shape encountered, selected to contain groups of closely matching shapes. The rejection and substitution rates achieved by the moments method when presented with the *limited training set* do not achieve the original specification (table 8.2). The results are grossly pessimistic since the test set contains many more closely matching shapes than would be found in a typical data base. The following explains the reason for this disproportionately high substitution rate:-

A component (class 1) represented by a prototype p1 has an approximately Gaussian distribution (this was determined experimentally - see Appendix VI) as shown in figure 8.1. The component is taught to the system using a single component p1. The region within which this component is assumed to exist is defined by a tolerance band either side of p1, i.e. p1 $\pm$ t. In recognition mode any class 1 component outside this region (i.e within the hatched region) will be rejected.

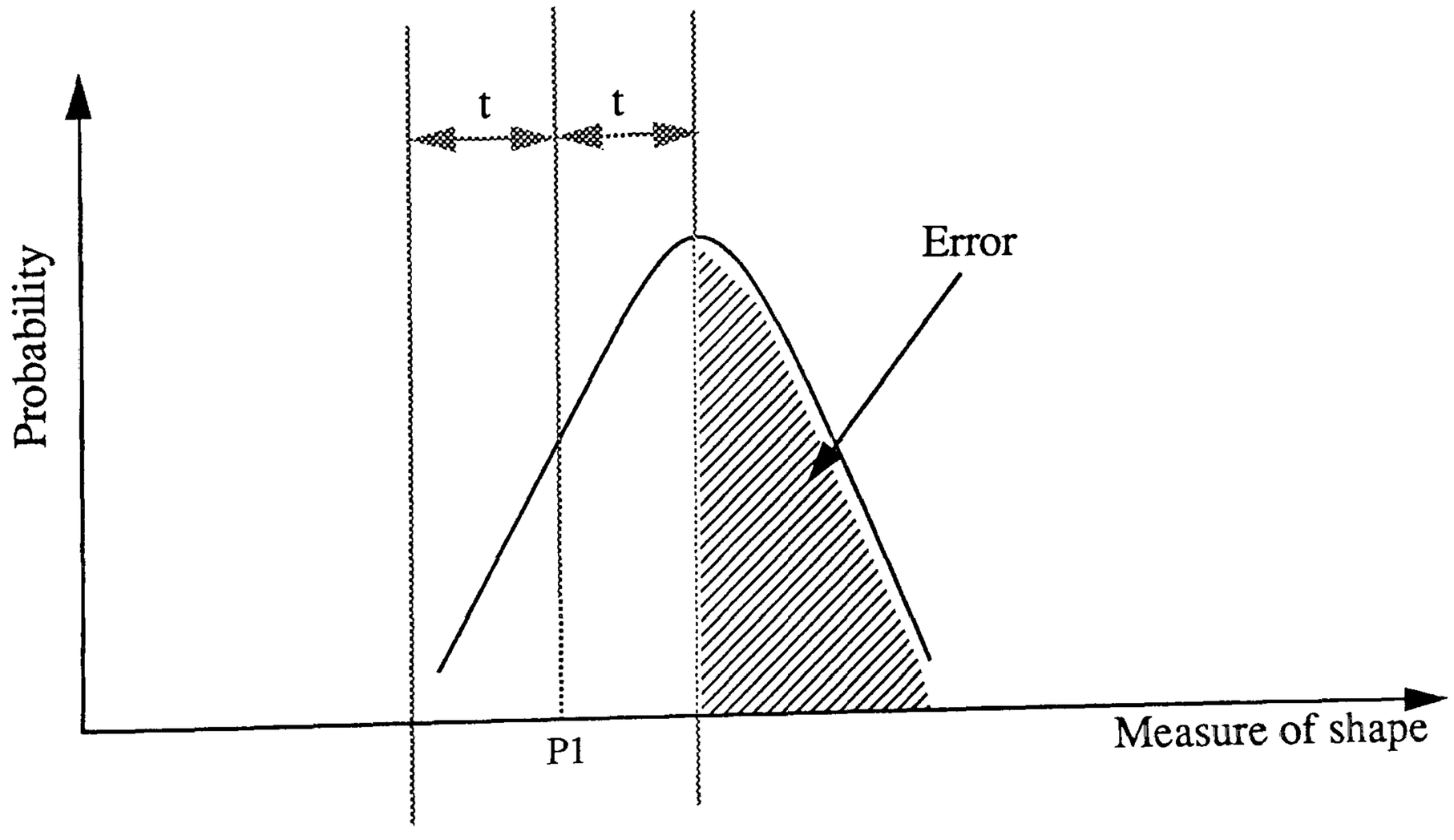


Figure 8.1 Error for a single isolated component.

If a second component (class 2) represented by prototype p2 and defined in the region p2 $\pm$ t exists close to the mean of class 1 components, then some of the rejected components will be substituted as shown in figure 8.2. The total error for component class 1 remains the same.

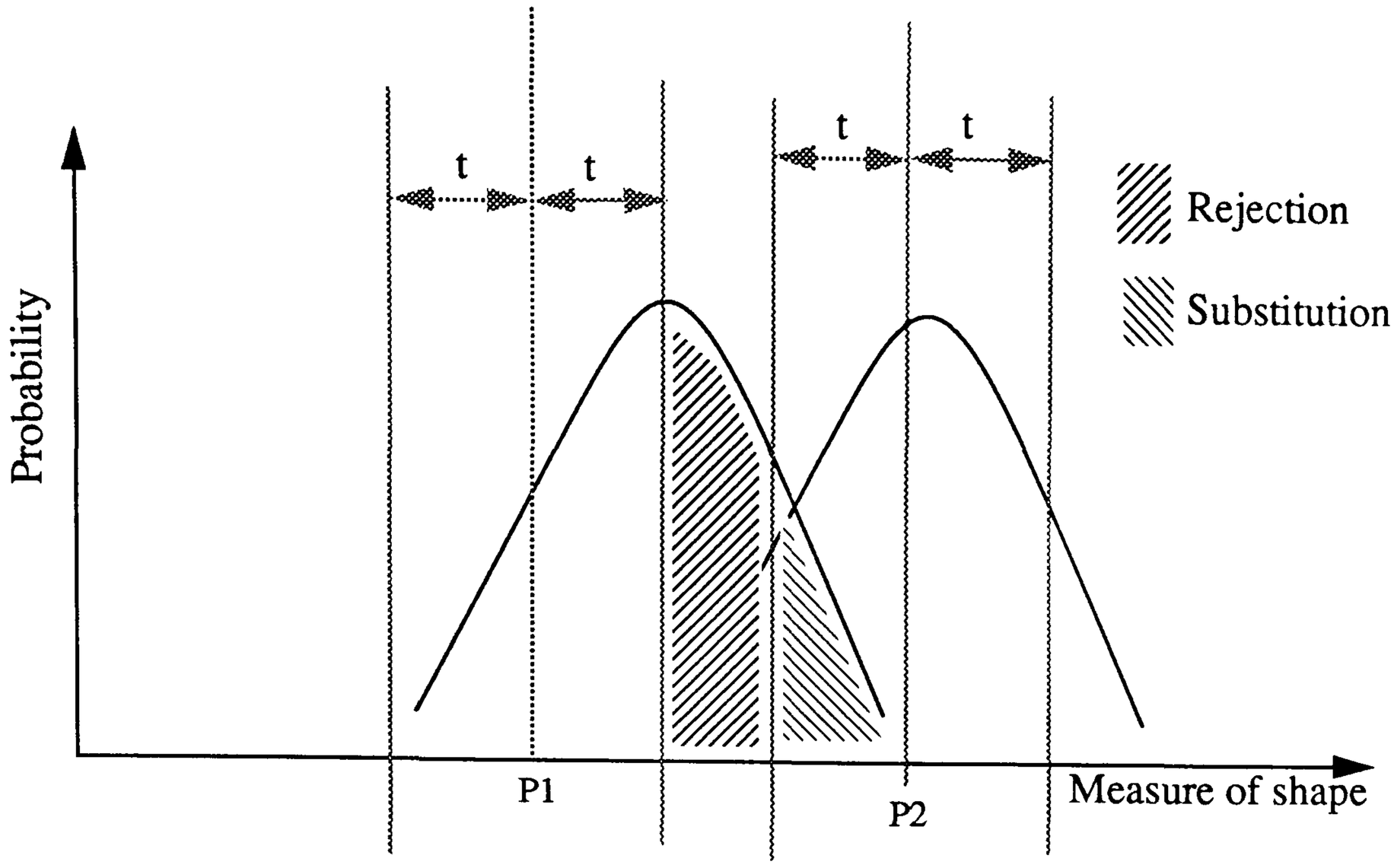


Figure 8.2 Error for two close shapes.

Cases will occur (e.g. Tsh16 and Tsh09) in which the regions assigned to class 1 and class 2 will overlap (figure 8.3) such that an unidentified component can appear in both regions (p1± t & p2 ± t.). Identification is then resolved by determining which prototype is the nearest. Components which would have been identified correctly in the situations defined by figures 8.1 and 8.2 might now be substituted. Thus the total error increases.

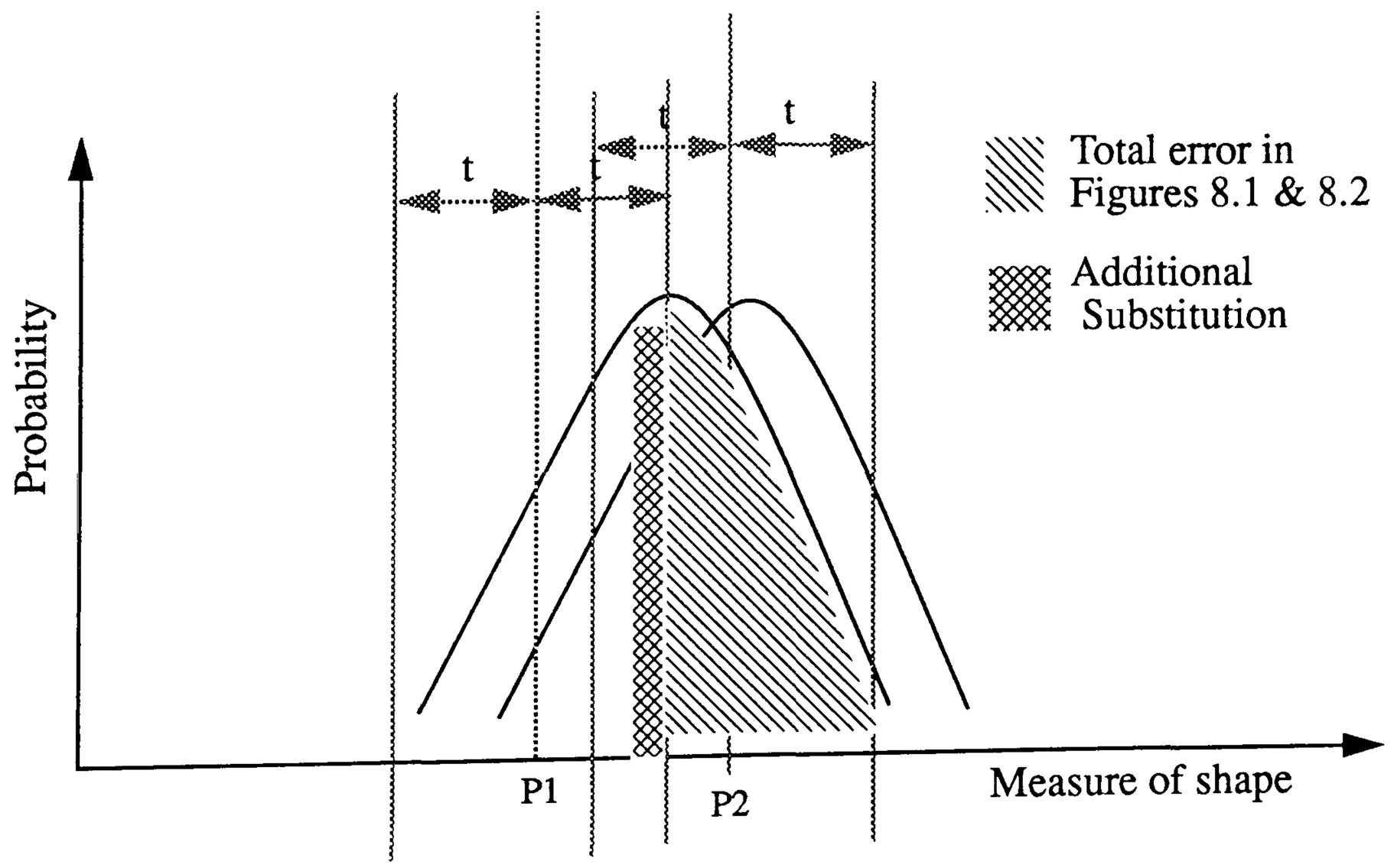


Figure 8.3 Error for two VERY close shapes.

Figure 8.4 illustrates the effect on the error (hatched areas in figures 8.1, 8.2 & 8.3) for any single component as the distance (e.g. the Euclidean distance) between a pair of adjacent components is varied.

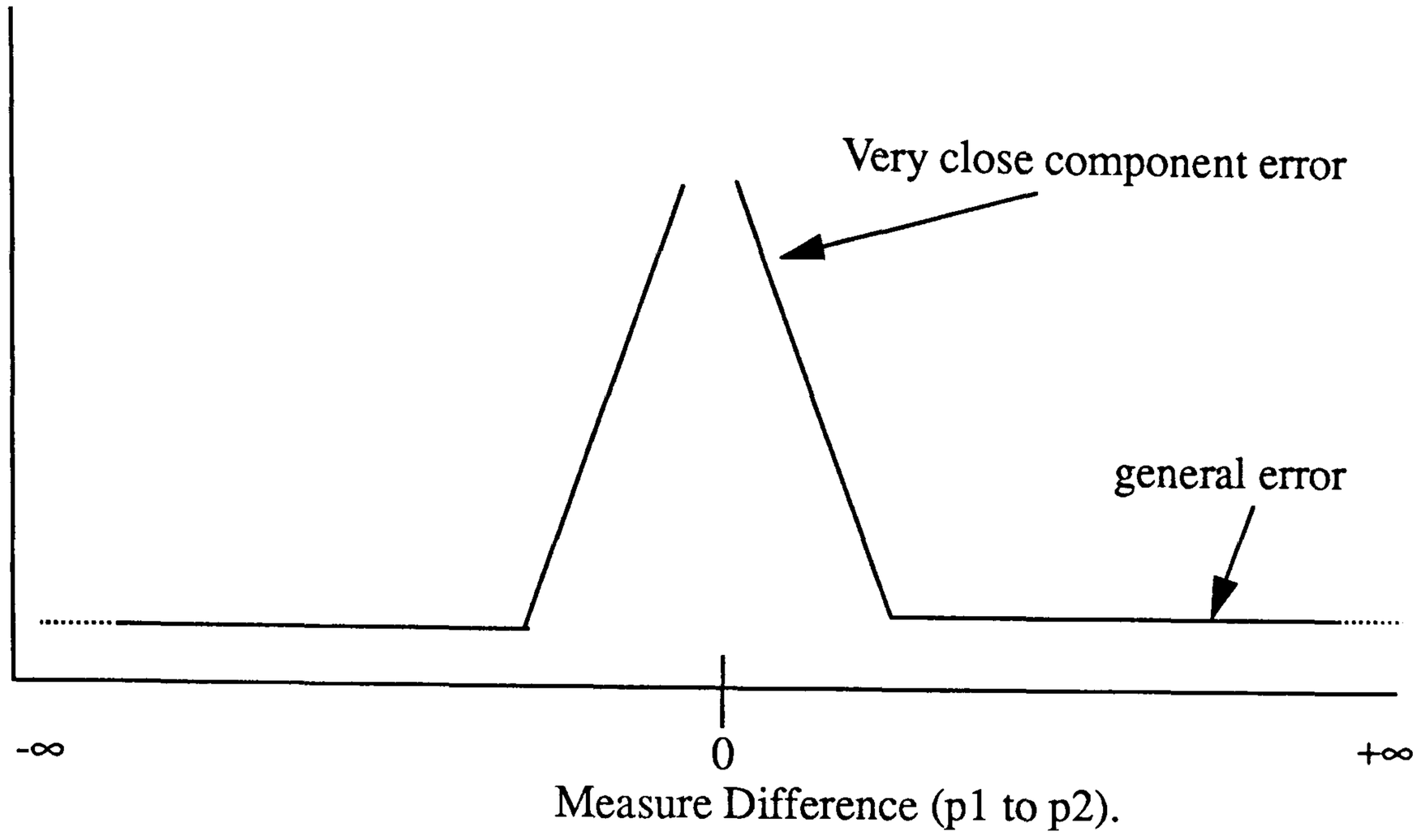


Figure 8.4 Failure rate variation with shape closeness.

The components in the test set are grouped together in families of very similar shapes. Thus the results contain many errors of the type explained here. The actual results in a factory situation would depend on the number of very close shapes present in the database. These are anticipated to be lower than in this test set and therefore to have a lesser effect, reducing the substitution rate.

8.5 Summary

The results presented in this chapter show that neural networks can be used to gain a significant advantage in recognition performance without compromising operational speed. This is achieved by only using the neural networks to identify very closely related shapes and relying on feature matching techniques to separate the majority of shapes. The time that would be required to train neural networks on all shapes in the database in thus avoided.

The recognition performance of the moment invariant based system using feature matching only was about 9% better than that of the radii to centroid method used by Tout. A further improvement over Touts method (33%) has been obtained by utilising neural networks. It is, however, reasonable to expect that the radii to centroid method would also benefit from the use of neural network techniques. The use of a feature algorithm which is readily compatible with the scanning system suggests that real increases in speed could be achieved.

CHAPTER 9

CONCLUSIONS, SUGGESTIONS FOR FURTHER WORK AND ALTERNATIVE APPLICATIONS

9.1 Introduction

This thesis has presented investigations into ways of improving the performance of a machine vision system required to identify shoe upper components. The system is also required to calculate the orientation and position of such components which are presented as silhouette shapes viewed by a linescan camera.

This chapter reviews the outcome of the research with respect to the 1979 specification [CROUCH 1979] and discusses the improvements over the methods presented by Koulopoulos [KOULOPOULOS 1982] and Tout [TOUT 1989].

The machine vision system is not limited to shoe upper recognition for the purposes of stitch marking and alternative uses are suggested both within the shoe industry and general manufacture.

Finally suggestions for improving the methodologies and new avenues of research are discussed.

9.2 Research conclusions

Koulopoulos concluded, from his research, that radii from the centroid to the edge of a silhouette shoe component were a satisfactory form of feature representation for such shapes. In reaching this conclusion he chose to ignore all methods which were not easily visualised and thus rejected the moment method investigated extensively in this thesis.

Tout researched a complete system for stitching patterns onto randomly presented shoe components. His work included some consideration of other identification methods but

he concluded that the radii from centroid method as proposed by Koulopoulos was satisfactory. Tout did not investigate the extensive use of moments. He did, however, enhance the recognition system by including the second-moments of each component as a feature. Tout also used the area, a very stable value, as an initial filter prior to any radii comparisons thus speeding up data base search times.

In addition to component identification Tout also required component position and orientation. The component position problem was adequately solved by the average coordinates of the silhouette pixels as defined in equations 6.3 and 6.4. The accurate determination of orientation proved a more difficult problem since the efficient method using equation 3.6 was only applicable to 75% of components, the remainder requiring a cumbersome approach utilising the matching of 360 additional radii.

9.2.1 Recognition

This work has shown that the moments of a silhouette image can be used to efficiently solve all the required machine vision problems. The moments of the image can be used to produce a set of features which are invariant to translation and rotation, thus identifying shoe upper components. Experiments have demonstrated that although noise introduces errors which increase with the order of the moments, closely similar patterns can still be distinguished. Unfortunately the success rate for distinguishing such shapes is not as good as the level originally specified by BUSM. The specification does not, however, specify any limit in terms of the similarity between distinguishable shapes. It must be accepted that there is a level at which shapes are no longer separable.

Three methods for reducing the effects of noise have been investigated:-

- For all components, an averaging filter is used to remove dust particles from the image.

- This is followed by the adjustment of the calculated features to compensate for the quantisation errors imposed by the imaging system.

- The separation of very close components can be achieved by applying the moment invariant features to a system of neural networks.

The initial filtering of the images did not significantly affect the overall recognition performance. This is due to a lack of removable fibres in the research atmosphere. It is anticipated that a real industrial situation would have significantly more dust and dirt in the environment.

9.2.2 Position

The centroid of the component can still be found using equations 6.3 and 6.4 which are the first order moments divided by the area (zeroth order moment).

9.2.3 Orientation

The use of equation 3.6 as used by Tout is still the simplest and most efficient method for the calculation of component orientation. It is usable for all shapes with a GCOEFF value in excess of 0.2.

The components with a GCOEFF value of 0.2 or less require a different approach. A significant improvement over the comparison of 360 radii as used by Tout has been obtained by evaluating the solution to the rotation of the third order moment μ_{30} as given in equation 6.16. This method requires no additional features to be found and can be calculated in 3% of the time. The results are more accurate than those obtained through equation 3.6 and could be used for all component patterns irrespective of GCOEFF value.

9.2.4 Efficiency

The use of only eight features is an improvement on the twenty two used by Tout, reducing database search times and storage requirements. Methods have also been proposed and validated (section 6.5) for optimising the efficiency of computation of the moments.

9.2.5 Comparison

The Autostitcher machine currently sold by BUSM, based on the radii method researched by Tout and Koulopoulos has demonstrated that machine vision can be used in the shoe industry. The work presented in this thesis, using moment based features with or without neural networks, has improved on the previous results and might further the use of machine vision in shoe automation.

The recognition performance obtainable using moment based features has been shown to be comparable with that using the present "radii from centroid" approach, whilst offering more efficient processing. Whether it will be incorporated into BUSM's future stitching or similar machines is a development decision, but the methods have all been tried and tested successfully on prototype hardware. One possibility is the combined

running of both the radii and moments systems in parallel to give increased discrimination ability to the overall system.

9.3 Further work

There are several reasons for continuing the work on this project. In addition to simply improving the recognition performance alternative methods can be used to reduce the cost of the current system without necessarily achieving better performance.

9.3.1 Grey images

The system used in this research viewed the components through an analogue grey scale camera. The signal was thresholded at the camera before being transmitted to the hardware and software. All further processing is performed on binary data only. The control over this threshold was limited and removed the option of accurately setting the threshold or performing filtering based on the grey scale image.

The use of a grey scale camera and appropriate interface would present many more options for filtering the data prior to feature extraction. If the original image is noisy then all subsequent operations will suffer reduced performance.

9.3.2 Interpolation

Jinabhai [JINABHAI 1992] predicted that increased resolution would improve recognition performance. The most obvious way to implement this would be through the use of higher resolution cameras with associated cost. Interpolation techniques would allow the increased resolution to be approximated without increased cost. Alternatively the current camera size could be reduced, giving a cost saving, without lowering system performance.

9.3.3 Parallel processing

The processing time of the system, in particular the data base searches, could be reduced by utilising the parallel processing attributes of transputers and similar processors.

9.3.4 Alternative features

Chapter 3 reviewed the methods of shape representation considered by Tout and Koulopoulos while methods considered in this research were presented in chapter 5. Other methods of shape representation could be considered, with a view to finding more robust features. This work ultimately chose moments as shape descriptors because they offered fast and efficient computation. Faster processors may make less efficient methods realistic.

9.3.5 Integrated production line

Within the shoe factory there are many processes which could be linked together such that a series of components fed in at one end arrive as finished shoes at the other. The decorative stitching of shoe upper components is one of the initial processes. The information obtained by the machine vision system could be used to control the manipulation and processing of the component on subsequent machines.

9.3.6 Neural networks

The neural networks used in this research have been employed in the traditional role of pattern recognition. Unfortunately the time handicap associated with training the networks will preclude their general use in a production line system. There are however alternatives methods and uses which might be investigated;

9.3.6.1 Parameter optimisation

Although moment invariants have been shown to be adequate recognition features the representation chosen is probably sub optimal. Iivarinen *et al.* [IIVARINEN *et al.* 1994] and De'Vena [DE'VENA 1994] have investigated the use of neural networks for optimising the feature selection. Such analytical methods could be applied to the moment invariant feature vector. In such circumstances on-line, real time performance would not be required.

9.3.6.2 Alternative methods

The Back-Propagation, feed forward algorithm used to train the neural networks is only one of many now published. Alternative training algorithms and network architectures

[e.g. CARPENTER 1989] might improve the recognition performance or be required to achieve the objectives in 9.3.6.1.

9.3.7 Similarity and substitution

As is to be expected, increased similarity between shapes leads to a greater level of substitution. This relationship has not been quantified and an investigation in this area would help to form a better understanding of the recognition machine.

9.4 Alternative applications

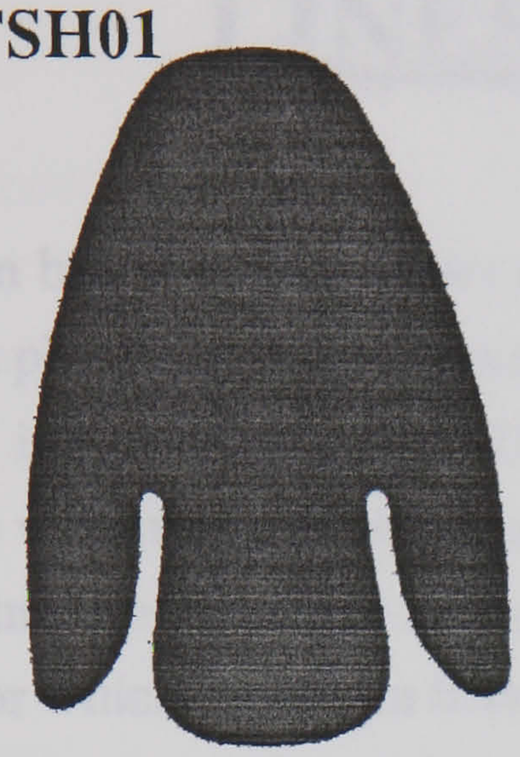
The work on the machine vision system has been specifically linked with BUSM's range of Autostitcher machines due to the nature of the CASE sponsorship agreement. The Autostitcher machine is designed to stitch decorative patterns onto shoe components prior to shoe assembly. This general approach of recognising two dimensional shapes and calculating their position is relevant to many operations both within BUSM and manufacture generally. Shoe components have many properties which make them difficult to identify and as such they provide a good basis for a general system.

The general approach of single teaching makes training times low. Thus this type of system is suited to tasks involving a high variety and low volumes of components where re-tooling occurs frequently. Due to its ability to cope with many different components it is also suited to processes involving a high volume of varied components.

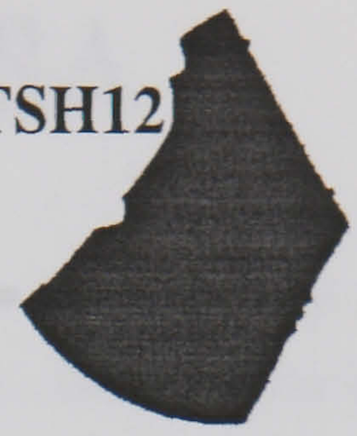
APPENDIX I

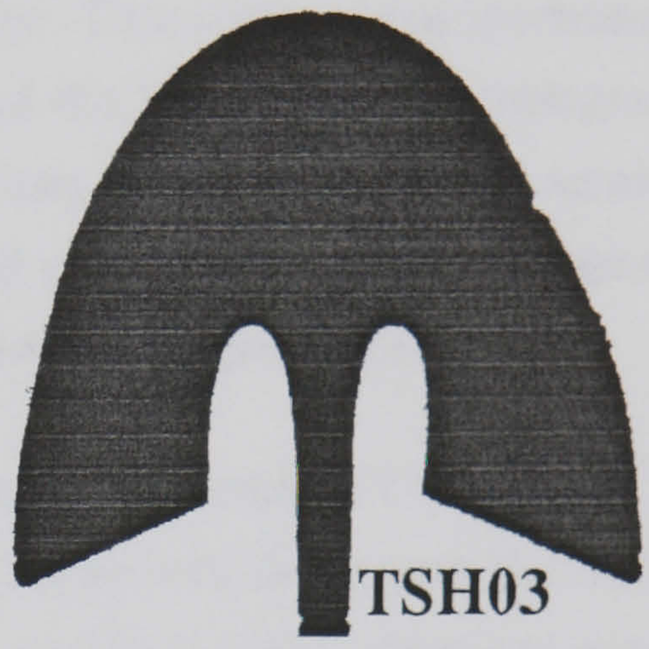
EXAMPLES OF SHOE COMPONENT SHAPES

TSH01



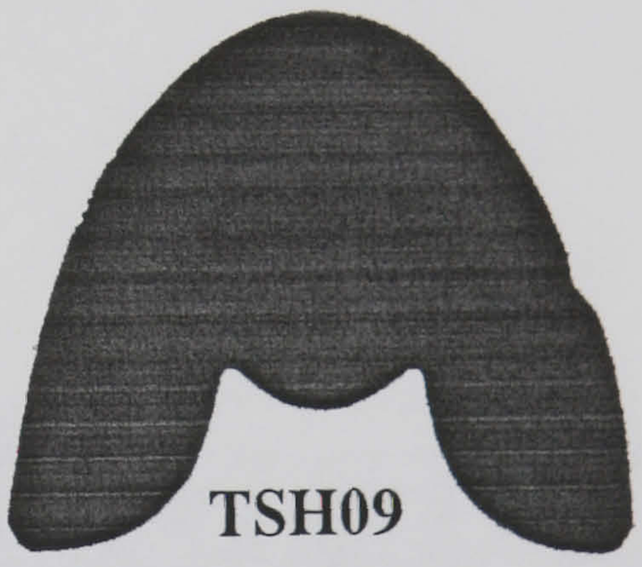
TSH12



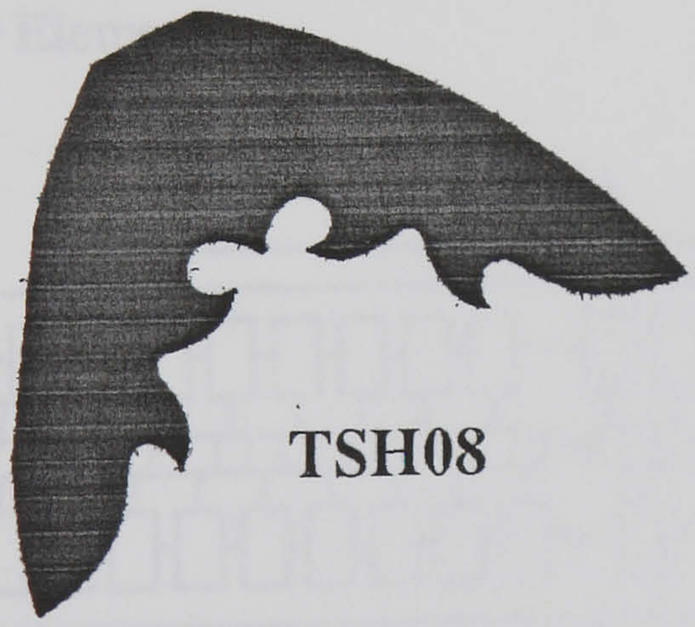


TSH03

TSH07

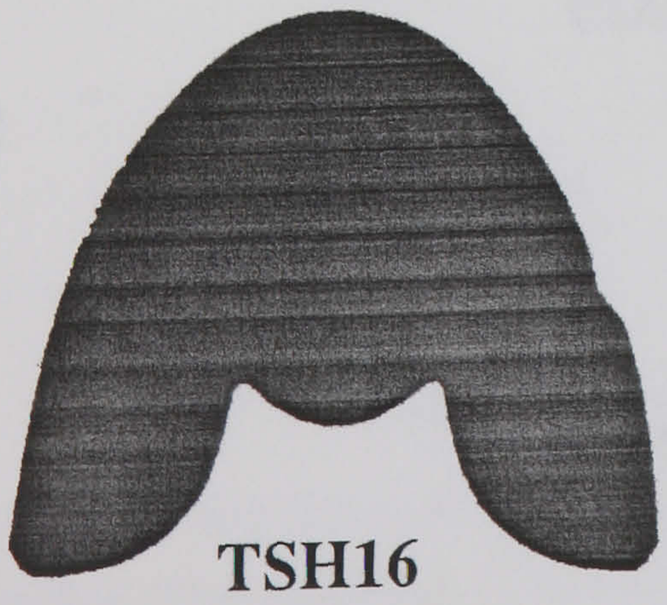


TSH09



TSH08

TSH04



TSH16

APPENDIX II

LINESCAN CAMERA

A linescan based system views an image one line at a time. In between each scan line the object plane and the camera must be moved relative to each other in order to build a complete image of a scene. During each scan the charge coupled devices (CCD elements) within the camera array will build up a charge. This charge is proportional to the brightness of the light falling onto the element and the length of time (integration period) for which it falls. In between successive scan lines the CCD array will receive a SYNChronisation pulse which will cause the collected charge in the CCD elements to be transferred to corresponding shift registers as detailed in Figure AII.1.

The scan line can then be passed to external hardware by serially CLOCKing and reading the shift registers. The length of the scanning array will determine the number of CLOCK pulses required to read out all of the data and hence the minimum number of CLOCK cycles between successive SYNC pulses.

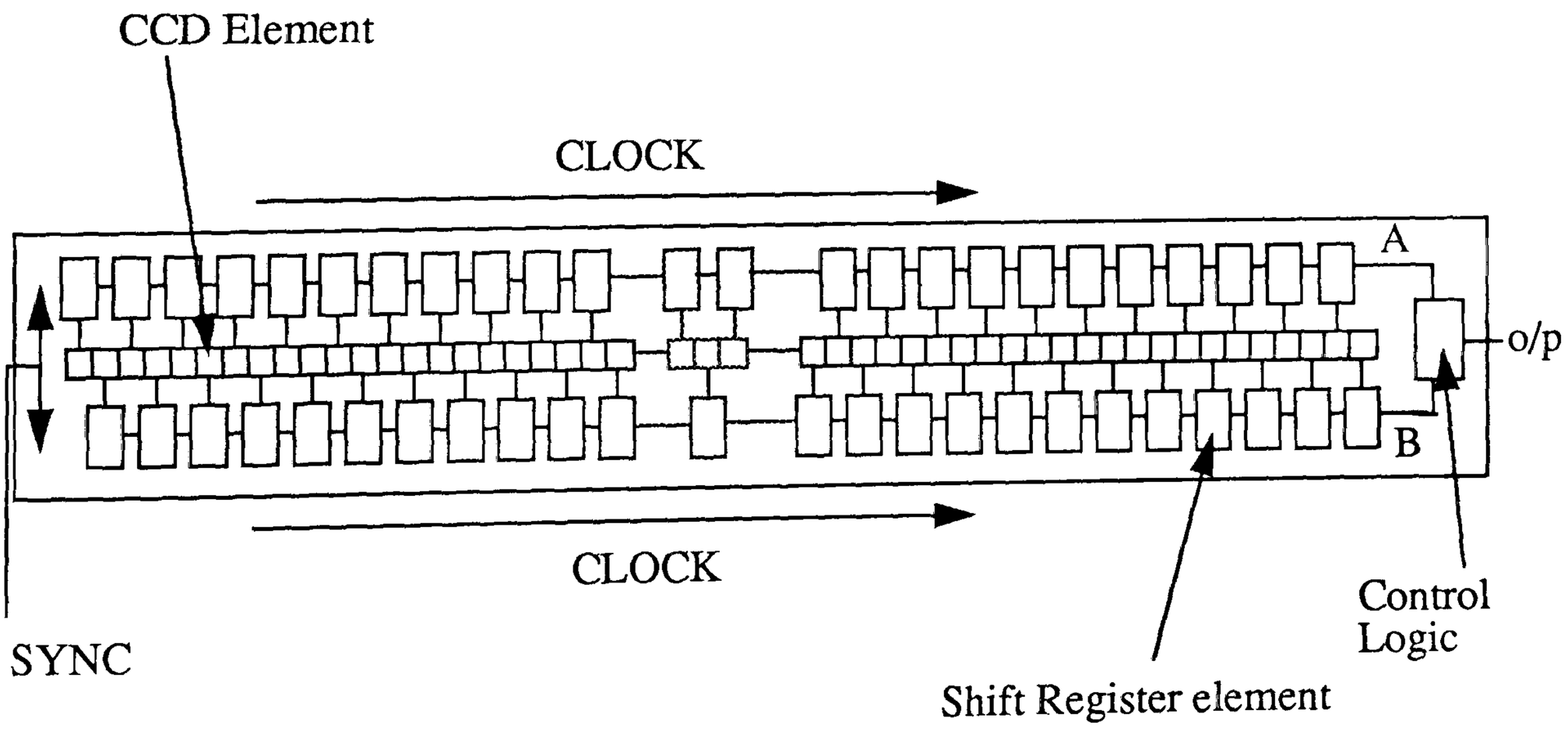


Figure AII.1 Linescan arrays.

APPENDIX III

SYSTEM HARDWARE AND COMMUNICATIONS

BUSM's new hardware is based on the use of stand alone modules, using the latest technology such as transputers and DSP's (Digital Signal Processors). The research reported in this thesis is required to perform on the vision module, the operation of which is described here.

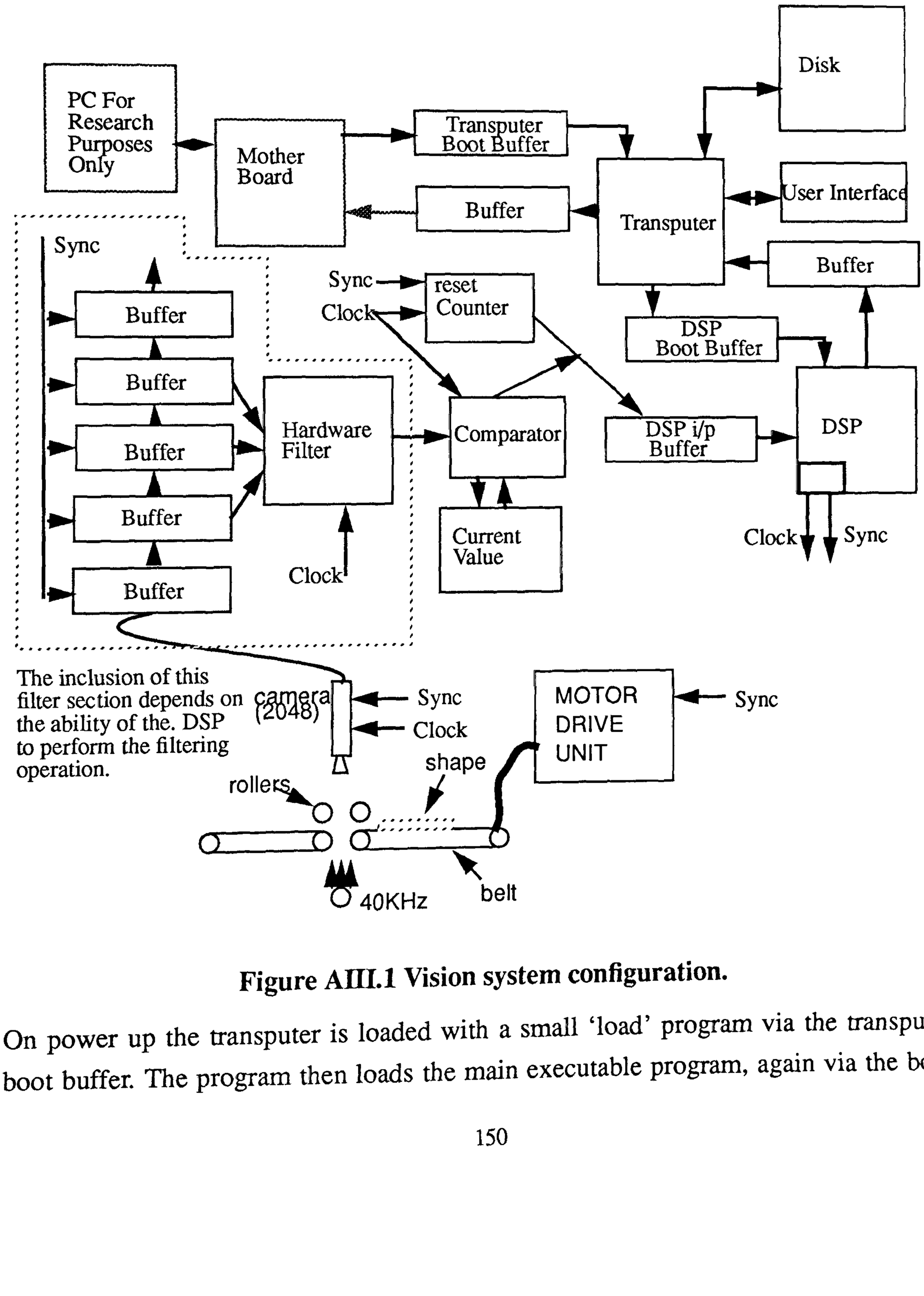


Figure AIII.1 Vision system configuration.

On power up the transputer is loaded with a small 'load' program via the transputer boot buffer. The program then loads the main executable program, again via the boot

buffer. The transputer places a similar small 'load' program into the DSP buffer which in turn loads the main DSP program. This cumbersome method of system initialisation is required since the internal boot programs of the DSP and Transputer were found to fail for larger amounts of data such as the main executable programs.

The internal DSP clock is monitored by an interrupt routine which generates the sync pulse every 2048 cycles. Figure AIII.2 shows the relationship between the clock and sync pulse. The sync pulse causes the camera to transfer any charge from the CCD array into its shift registers and corresponds to a 'new line' instruction. Note that the DSP clock is suppressed during the sync pulse so that no data is transferred into the hardware during the charge transfer. The 2048 clock pulses in between any two sync pulses are used to move the data pixels serially out of the camera and into the hardware. The pixel values in the camera shift registers are analogue charge values. As the pixels are read from the shift registers they are converted by the camera into voltage signals, thresholded and inverted. The input to the hardware is thus a binary signal with light pixels represented by 0 volts and dark pixels by 1 volt. Each binarised pixel value is stored on the board in one of five shift registers corresponding to the current line and the four preceding lines. The sync pulse is also used to reset the comparator clock and to advance the component being scanned by one line (0.2 mm).

The middle shift register is processed by the filter hardware which executes equation 6.53 (section 6.6.1) and the filtered line fed into the comparator. If the DSP is able to perform the filtering operation with adequate speed then the data from the camera will be fed straight into the comparator. The DSP will then filter the compressed data prior to extracting the moment values for each line.

The comparator compares each pixel with that which came before. If any difference is found then the x address of that pixel as determined by the counter is sent to the DSP's input buffer. The counter is reset with every sync pulse and incremented by the clock pulse so its value corresponds to a pixel's x address.

The DSP reads in the addresses from the buffer and calculates the moments, relative to the intersection of the y axis (pixel number 0) and the first scan line of the component, as described in section 6.5.1. When the end of the shape is detected due to two consecutive addresses being outside the operating window the DSP will calculate the centroid, the transitional moments and the invariant moments as described in section 6.5.2. The calculated values are sent to the transputers input buffer.

The task performed by the transputer will depend on whether the system is in learn or recognition mode. In learn mode the features, position and orientation of the prototype

shape will be written to disk. In recognition mode the transputer will search the disk database and calculate the components identity. In either case the components name, orientation and position would be fed to the system mother board which would then activate the sewing machine module or similar process.

The use of buffers between the DSP / transputer and Hardware / DSP allows the processors to work asynchronously of the data arrival. Ideally each line of scan will be processed by the DSP in the period between pairs of sync signals. By using buffers the DSP can lag behind the camera scans if any complex lines are encountered.

From a functionality point of view the board can be split into three areas; the image processing hardware; the feature extraction software and the component identification software. Only the image processing hardware uses the raw image data. Both of the software functions use a compressed version of the image formed from selected data points.

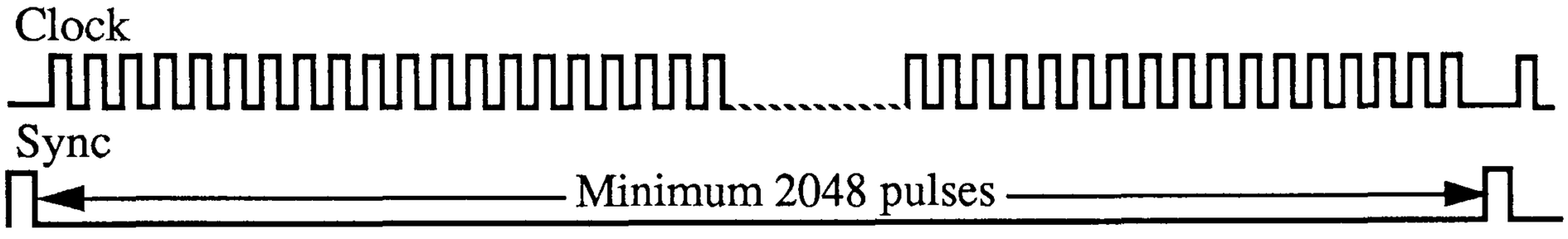


Figure AIII.2 Relationship between CLOCK an SYNC signals.

APPENDIX IV

COMPONENTS AND COMPONENT GROUPINGS

GROUP 1:-

TSH01, TSH38.

GROUP 2:-

TSH02

GROUP 3:-

TSH05

GROUP 4:-

TSH03, TSH06, TSH28.

GROUP 5:-

TSH08

GROUP 6:-

TSH09, TSH10, TSH16, TSH19.

GROUP 7:-

TSH11, TSH12, TSH13, TSH14, TSH15.

GROUP 8:-

TSH20, TSH21, TSH23.

APPENDIX V

ERROR CORRECTION CURVE PARAMETERS

FEATURE 0	FEATURE 1	FEATURE 2	FEATURE 3
a=2.126968e-02	a=0.1423024	a=0.1823895	a=0.3558381
b=0.0277729	b=2.846661e-02	b=2.883246e-02	b=2.939372e-02
c=-2.828824e-06	c=-5.173748e-05	c=-6.802724e-05	c=-1.376435e-04
d=-3.022293e-06	d=-3.522121e-06	d=-3.715985e-06	d=-4.350381e-06
λ=0.1519	λ=0.3827473	λ=0.72363422	λ=1.281592
γ=0.2792305	γ= 0.2740024	γ=-1.115106	γ=0.2485697

FEATURE 4	FEATURE 5	FEATURE 6	FEATURE 7
a=9.519276e-02	a=-0.1589081	a=-0.4499074	a=0.3456332
b=0.0313322	b=2.864199e-02	b=2.506592e-02	b=3.022917e-02
c=-3.91577e-05	c=5.929062e-05	c=1.612696e-04	c=-1.558338e-04
d=-4.315889e-06	d=-3.792995e-06	d=-3.164447e-06	d=-4.705248e-06
λ=5.51018	λ=5.51018	λ=3.567646	λ=15.31949
γ= 0.643275	γ= 0.643275	γ= -0.1034339	γ= -0.8872646

APPENDIX VI

COMPONENT FEATURE DISTRIBUTION

The feature values for a type of component will vary with the position and orientation of a shape. There will also be a variation between nominally identical examples of a shape due to production variations and noise. The following results illustrate that the values of these features appear to be distributed in an approximately Gaussian fashion.

Tables VI.1 shows the number of occurrences of each region of error around the mean feature value for given sets of nominally identical shapes. The error for a feature value with respect to the mean of the available component examples is given in equation VI.1.

$$Error = \frac{F_x - \bar{F}_x}{\bar{\bar{F}}_x} \times 100\% \ldots\ldots\ldots\ldots \text{eq VI.1}$$

The results in table VI.1 are plotted in figure VI.1 and show that the error is tending towards a Gaussian distribution. The values used followed filtering of the image with a 5 by 5 pixel filter (section 6.6.1) and compensation for the image grid (section 6.6.2).

Error Range	Frequency
+7.75≤error	4
+7.25≤error<+7.75	0
+6.75≤error<+7.25	1
+6.25≤error<+6.75	1
+5.75≤error<+6.25	2
+5.25≤error<+5.75	2
+4.75≤error<+5.25	0
+4.25≤error<+4.75	1
+3.75≤error<+4.25	9
+3.25≤error<+3.75	12
+2.75≤error<+3.25	7
+2.25≤error<+2.75	10
1.75≤error<+2.25	18
+1.25≤error<+1.75	36
0.75≤error<+1.25	53
+0.25≤error<+0.75	101
–0.25≤error<+0.25	110

Error Range	Frequency
–0.75≤error<-0.25	98
–1.25≤error<-0.75	39
–1.75≤error<-1.25	30
–2.25≤error<-1.75	24
–2.75≤error<-2.25	9
–3.25≤error<-2.75	10
–3.75≤error<-3.25	10
–4.25≤error<-3.75	4
–4.75≤error<-4.25	7
-5.25≤error<-4.75	5
-5.75≤error<-5.25	4
-6.25≤error<-5.75	3
-6.75≤error<-6.25	1
-7.25≤error<-6.75	1
-7.75≤error<-7.25	1
error<-7.5	1

Table VI.1 Occurrences of Features in given error range.

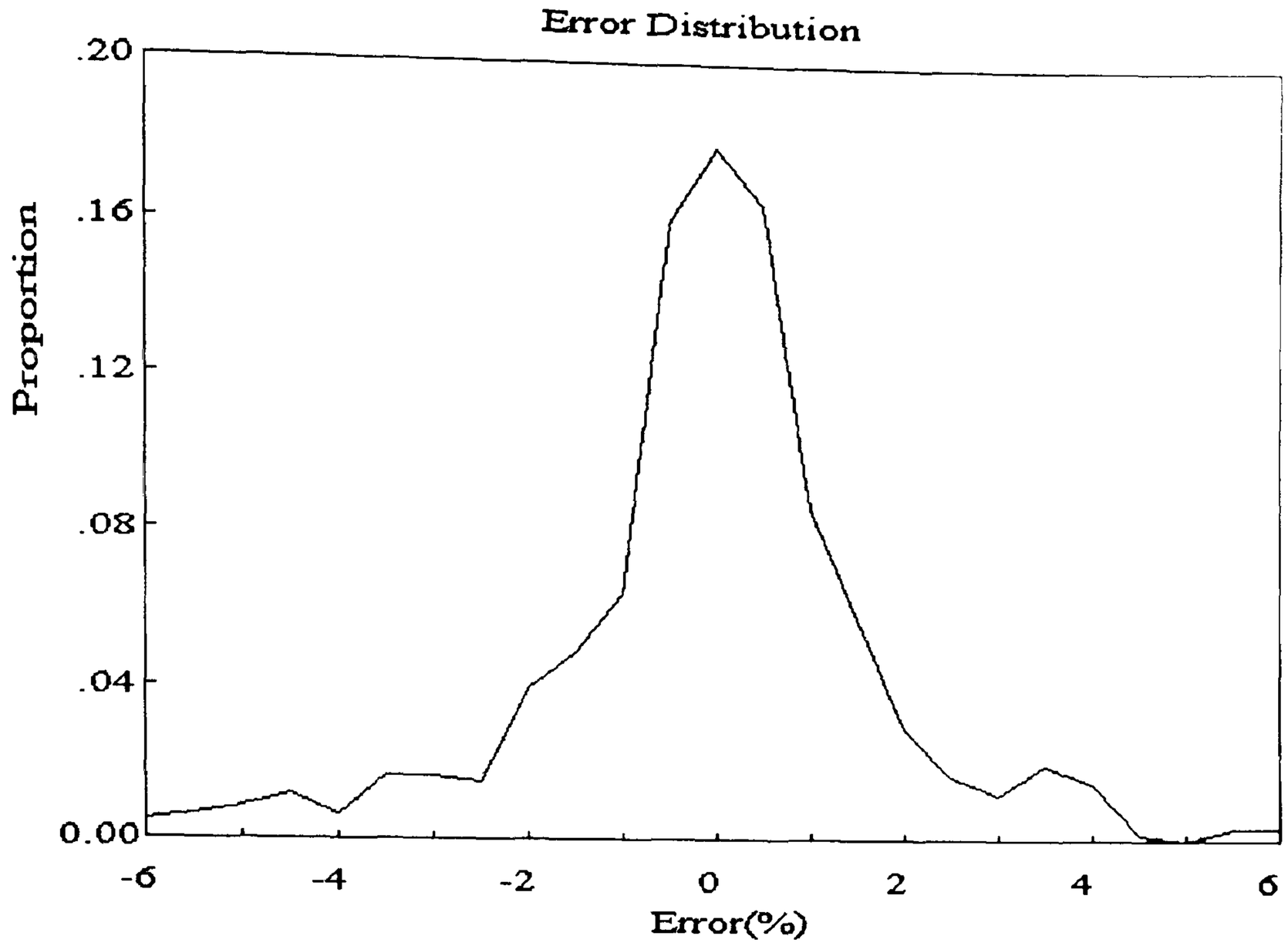


Figure VI.1 Proportion of given errors occurring.

If the distribution is assumed to be Gaussian (the samples have a mean of -0.0318 and a variance of 3.93) then it can be presented as in figure VI.2 and evaluated using the normal distribution [e.g. CHOU 1969].

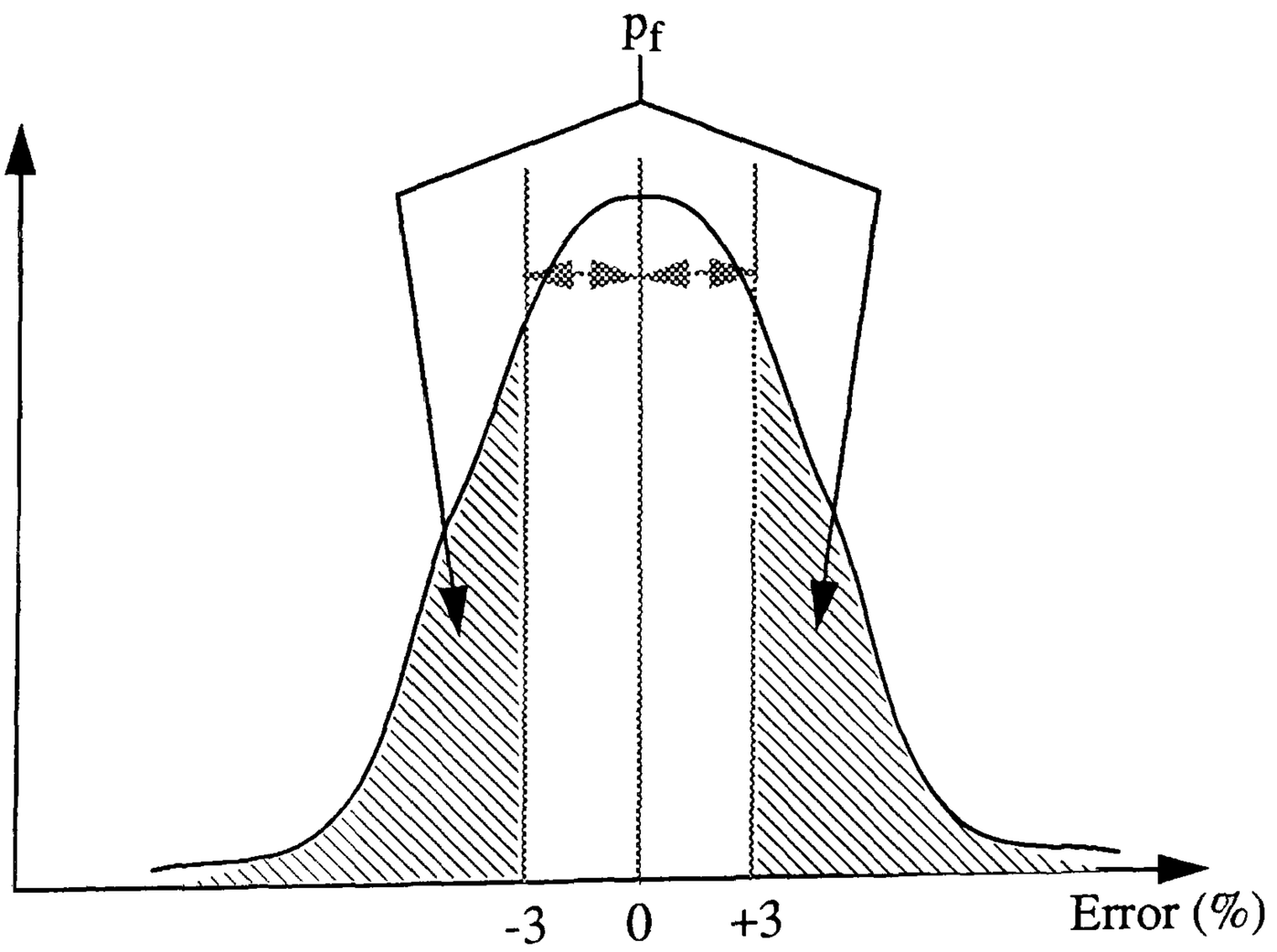


Figure VI.2 Assumed normal distribution of features.

If it is assumed that all prototypes lie on the mean value then the probability of failure for feature f_x is given by the area p_f. P_f can be evaluated using normal distribution data [e.g. CHOU 1969]. For the distribution used in figure VI.2 p_f is 0.126. Repeating this process for all eight features gives an average value for p_f of 0.0928. Referring to equation 4.9 (section 4.5.1) and letting p(0) = the average value for p_f then the general failure rate is estimated at 0.382%. Section 8.4 shows why this value will be an underestimate due to the assumption that all prototypes have mean values.

The estimation of the value of p(0) will vary with the distribution of the representative features.

REFERENCES

ALEKSANDER, I. & MORTON, H. (1990). An Introduction to Neural Computing, Chapman and Hall, London.

BALLARD, D. H. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes, Pattern Recog., Vol. 13, pp 111-122.

BEALE, R. & JACKSON, T. (1994). Neural Computing - an Introduction, Institute of Physics Publishing, London.

BLOWER, N. (1982). Computer Simulation Using Ellipses, Rectangles and Triangles to Find Orientation Using Moments of Area, BUSM Internal Calculation Sheet.

BLUMENKRANS, A. (1991). Two-Dimensional Object Recognition Using a Two-Dimensional Polar Transform, Pattern Recog., Vol. 24, No. 9, pp 879-890.

BORN, M. & WOLF, E. (1970). Principles of Optics, Pergamon Press, London.

BOON, J. EVANS, D. HENNIGAR, H. (1982). Spectral Information from Fourier Analysis of Digitized Quartz Grain Profiles, Math. Geol., Vol. 14, No. 6, pp 589-605.

BREZINA, J. (1977). Radius Density Method, Univ. London Comp. Centre, Seminar.

BROWNE, A. & NORTON-WAYNE, L. (1986). Vision and Information Processing for Automation, Plenum Press, London.

BURKE, H. E. (1984). Handbook of Bar Coding Systems, Van Nostrand Reinhold Co. Ltd.,Wokingham.

CARPENTER, G. A. (1989). Neural Network Models for Pattern Recognition and Associative Memory, Neural Networks, Vol. 2, pp243-257.

CASE, J. & CHILVER, A. H. (1971). Strength of Materials and Structures, Edward Arnold Ltd., London.

CHARTIER, P. (1994). 2-D Encodable Symbologies, STEP Lecturer Institute Seminar, Keele Univ.,

CHESTER, W. (1979). Mechanics, George Allen & Unwin Ltd., London.

CHOU, Y. (1969). Statistical Analysis with business and economic applications, Holt, Rinehart & Winston, London.

CROUCH, I. P. (1979). Shape Recognition of Shoe Uppers, Internal letter BUSM.

DAVIES, E. R. (1990). Machine Vision, Theory, Algorithms, Practicalities, Academic Press, London.

DE'VENA. L. (1994). Automatic Selection of the Most Relevant Features to Recognize Objects, Proc. of the Int. Conf. on Artificial Neural Networks 1994, Sorrento, Vol. 2, pp 1113-1116.

DUDANI, A. BREEDING, K. J. McGHEE, R. B. (1977). Aircraft Identification by Moment Invariants, IEEE. Trans. Compt., C-26, pp39-45.

EVANS, A. (1993). Mix and Match, Image Proc., Vol. 5, Issue 3, pp 17- 20.

FREEMAN, H. (1962). On the Digital Computer Classification of Geometric Line Patterns, Proc. Nat. I. Electronics Conf., Vol. 18, pp 312-324.

GALBIATI, L. J. (1990). Machine Vision and Digital Image Processing Fundamentals, Prentice-Hall International (U.K.) Ltd., London.

GALLAGER, R. G. (1968). Information Theory and Reliable Communications, John Wiley & Sons, Chichester.

HECHT-NIELSEN, R. (1991). Neurocomputing, Addison-Wesley Pub. Co., Wokingham.

HU, M. (1961). Visual Pattern Recognition by Moment Invariants, IRE trans. on Inf. Theory, IT-8, pp 179-187.

HUGHES, G. F. (1968). On the Mean Accuracy of Statistical Pattern Recognisers, IEEE. Trans. in Recog. Theory, Vol. 14, pp 55-63.

HUSSAIN, Z. (1991). Digital Image Processing, Ellis Horwood Ltd., London.

IIVARINEN, J. VALKEALAHTI, K. VISA, A. SIMULA, O. (1994). Feature Selection with Self-Organizing Feature Map, Proc. of the Int. Conf. on Artificial Neural Networks 1994, Sorrento, Vol. 1, pp 334-337.

JAIN, R. KASTURI, R. SCHUNCK, B. G. (1995). Machine Vision, McGraw-Hill, London.

JINABHAI, D. (1992). Analysis and Improvement of BUSM Recognition System. M.Sc. Report, De Montfort Univ., Leicester.

KANE, R. & MILGRAM, M. (1994). Forecasting Using Constrained Neural Networks, Proc. of the Int. Conf. on Artificial Neural Networks 1994, Sorrento, Vol. 1, pp 284-287.

KOULOPOULOS, C. (1982). Automated Vision for Identification of Silhouette Shapes with an Application to Industrial Automation, Ph.D. Thesis, City University, London.

LANITIS, A. Reading Hand Printed Postcodes Using Flexible Template Matching, Manchester Univ., BMVA, 1000 word extended abstract from student meeting.

LEAVERS, V. F. (1992). Shape Detection in Computer Vision Using the Hough Transform, Spinger-Verlag, London.

LI, Y. (1995). Optical Control of a Commercial Knife Bending Machine, Ph.D. Thesis, Bradford University.

McCABE, D. NORTON-WAYNE, L. BOLTON, G. SKIPPER, A. M. (1994). A Camera Vision Control System for the Manufacture of Rivets, Proc. IEE Coll. on the Des. Implementation and use of Obj. Ori. Sys., London, pp 7/1-7/10.

MILLER, R. G. (1989). Manual of Shoemaking, Clarks Ltd.

NAHIN, P. (1972). Slope Density Method, IEEE Comput., Vol. 21, pp 1233.

NIBLACK, W. (1986). An Introduction to Digital Image Processing, Prentice-Hall, London.

NORTON-WAYNE, L. (1982). A Coding Approach To Pattern Recognition, Pattern Recog. Theory and Applications, D. Reidel Publishing Co., pp 93-102.

NORTON-WAYNE, L. (1994). The Efficient Analysis of Silhouette Images, Bulletin IMA, Vol. 30, No. 5/6, May/June, pp 89-90.

NORTON-WAYNE, L. & HUTTON, G. (1985). Intelligent Instrumentation for the Automated Inspection of Tacks, Proc. 2nd Intl. Conf. of Products and Services for the Automated Factory, pp 155-164.

PETROSINO, A. & SALVI, G. (1994). Recognizing Planar Objects with Neural Networks, Proc. of the Int. Conf. on Artificial Neural Networks 1994, Sorrento, Vol. 2. pp 835-838.

PIPER, D. (1970). Data Processing in Biology and Geology, edited by Cutbill, J.. pp 97-103.

RITTER, H. MARTINETZ, T. SCHULTEN, K. (1992). Neural Computation and Self-Organizing Maps, Addison-Wesley Pub. Co., Wokingham.

SANBY, C. NORTON-WAYNE, L. HARWOOD, R. (1995). The Automated Inspection of Lace using Machine Vision, Mechatronics, Vol. 5, No. 2/3, pp 215-231.

SMITH, D. C. (1991). The Manipulation of Leather Workpieces for the Assembly of Shoe Uppers, Ph.D. Thesis, Hull University.

SMITH, F. & WRIGHT, M. (1971). Automatic Ship Photo Interpretation by the Method of Moments, IEEE Trans. on computers, Vol. 208 (Sept.), pp1089-1095.

TEAGUE, M. (1980). Image Analysis Via the General Theory of Moments, J. Opt. Soc. Am., Vol. 70, pp 920-930.

TOUT, N. (1989). Investigation of the Processes Required for the Automation of Stitchmarking in Shoe Manufacture, Ph.D. Thesis, Durham University.

WALI, R. COLEF, M. BARBA, J. (1991). Best Fit Ellipse for Cell Shape Analysis, SPIE, Vol. 1606, Visual Comms. and Image Proc. 1991, pp 665-674.

WASSERMANN, P. (1989). Neural Computing Theory and Practice, Van Nostrand Reinhold, New York.

ZAHN, C., & RUSKIES, R. Z. (1974). Fourier Descriptors for Plane Closed Curves, IEEE Trans. Computers, pp 843-849.

ZIGMANN, R. & SAULNIER, N. (1993). A Neural Network Application for Classification of Profiles, Comp. Vision for Ind., Munich 1993, SPIE. Vol. 1989, pp 267-274.